NASA
N-03-CR
166946
68P

DOT/FAA/CT-86/30

Quadruplex Digital Flight Control System Assessment

FAA TECHNICAL CENTER
Atlantic City International Airport
New Jersey 08405

D.B. Mulcare
L.E. Downing
M.K. Smith

Lockheed-Georgia Company
A Division of Lockheed Corporation
Marietta, Georgia 30063

July 1988 (Revised)

Final Report

This document is available to the U.S. public through the National Technical Information Service, Springfield, Virginia 22161.

US Department of Transportation
Federal Aviation Administration

(NASA-CR-182741) QUADRUPLEX DIGITAL FLIGHT CONTROL SYSTEM ASSESSMENT Final Report (Lockheed-Georgia Co.) 68 p CSCL 01C

N89-10058

Unclas
G3/08 0166946

NOTICE

This document is disseminated under the sponsorship of the U. S. Department of Transportation in the interest of information exchange. The United States Government assumes no liability for the contents or use thereof.

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the objective of this report.

Technical Report Documentation Page

1. Report No.
DOT/FAA/CT-86/30

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
QUADRUPLEX DIGITAL FLIGHT CONTROL SYSTEM ASSESSMENT

5. Report Date
July 1988 (Revised)

6. Performing Organization Code

7. Author(s)
D. B. Mulcare, L. E. Downing, M. K. Smith

8. Performing Organization Report No.
DOT/FAA/CT-86/30

9. Performing Organization Name and Address
Lockheed-Georgia Company
Marietta, Georgia 30063

10. Work Unit No. (TRAIS)
NAS2-11853

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
U.S. Department of Transportation
Federal Aviation Administration Technical Center
Atlantic City International Airport, New Jersey 08405

13. Type of Report and Period Covered
Contractor Report

14. Sponsoring Agency Code

15. Supplementary Notes
Point of Contact:

W. E. Larsen/MS 210-2
NASA/Ames Research Center
Moffett Field, CA 94035

Pete Saraceni, ACT-340
FAA Technical Center
Atlantic City International
Airport, NJ 08405

16. Abstract

This report describes the development and validation of a double fail-operational digital flight control system architecture for critical pitch axis functions. Architectural tradeoffs are assessed, system simulator modifications are described, and demonstration testing results are critiqued. Assessment tools and their application are also illustrated. Ultimately, the vital role of system simulation, tailored to digital mechanization attributes, is shown to be essential to validating the airworthiness of full-time critical functions such as augmented fly-by-wire systems for relaxed static stability airplanes.

17. Key Words
Design Verification, Digital Flight Controls, Fault Tolerance, Redundancy Management, Relaxed Static Stability, System Reliability, System Simulator, Validation Methods

18. Distribution Statement
Document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
70

22. Price

Reproduction of completed page authorized

FOREWORD

This report describes the rationale, objectives, procedures, and results of the airworthiness assurance process for fault-tolerant aspects of a quadruplex digital flight control system. Conducted as the basic task under NAS2-11853, the effort focused on critical pitch-axis functions for a relaxed static stability transport. Variations in redundancy management schemes were examined analytically, and considerable simulator testing was performed for the baseline system at the Reconfigurable Digital Flight Control System (RDFCS) Simulator at NASA Ames Research Center.

The intent of this project was to explore system architectures and associated assurance issues for critical functions such as stability augmentation accommodating negative static margins and fly-by-wire primary flight control. An integrated assurance approach that closely couples testing with analysis was employed, in a manner exemplifying key aspects of compliance with FAA Advisory Circular 25.1309-01. Both the investigations and this report were developed with the view toward its use for tutorial purposes.

TABLE OF CONTENTS

PAGE ii INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AC	Advisory Circular
AED	Algol Extended for Design (Programming Language)
AFBW	Augmented Fly-by-Wire
AOA	Angle-of-Attack
ARC	Ames Research Center
A/D	Analog-to-Digital (Converter)
C	C-Bar, or the Mean Aerodynamic Chord
CAPS	Collins Adaptive Processor System
DOF	Degree(s) of Freedom
DFCS	Digital Flight Control Systems
D/A	Digital-to-Analog (Converter)
EHV	Electrohydraulic Valve
EMAS	Electromechanical Actuation System
FAA	Federal Aviation Administration
FBW	Fly-by-Wire
FCC	Flight Control Computer
FSM	Finite State Machine
HOL	Higher-Order Language
Hz	Hertz, or Cycles per Second
IDEA	Integrated Digital/Electric Aircraft
I/O	Input/Output
IR&D	Independent Research and Development
MAC	Mean Aerodynamic Chord
MDICU	Modular Digital Interface Conversion Unit
MUX	Multiplex (Bus)
NASA	National Aeronautics and Space Administration
RDFCS	Reconfigurable Digital Flight Control System
RSS	Relaxed Static Stability
SAS	Stability Augmentation System
V&V	Verification and Validation

EXECUTIVE SUMMARY

A double fail-operational digital flight control system (DFCS) as shown in Figure E-1 was designed, analyzed, implemented, and validated relative to system fault tolerance and a subset of pitch-axis control functions. The system-level fault tolerance design, which from the outset was constrained by the Reconfigurable Digital Flight Control System (RDFCS) configuration at NASA Ames (Reference 1), was verified using a predicate/transition network simulation tool developed by the Lockheed-Georgia Company (Reference 2). The Ada (tm) programming language was used for software design, but the actual demonstration flight software was rendered in AED (Algol Extended for Design) as necessary for use in the RDFCS.

The demonstration at NASA Ames involved modifications to the original DFCS and system simulator as portrayed in Figure E-2. Basically, the dual-dual architecture was transformed into a quadruplex architecture through strictly software changes to extend fault tolerance for full-time flight criticality. Although the resultant implementation was sub-optimal from a real-time standpoint, it did realize the verified design. This served to illustrate the propagation and confirmation of consistency throughout the development process and to permit the demonstration of new real-time validation testing methods.

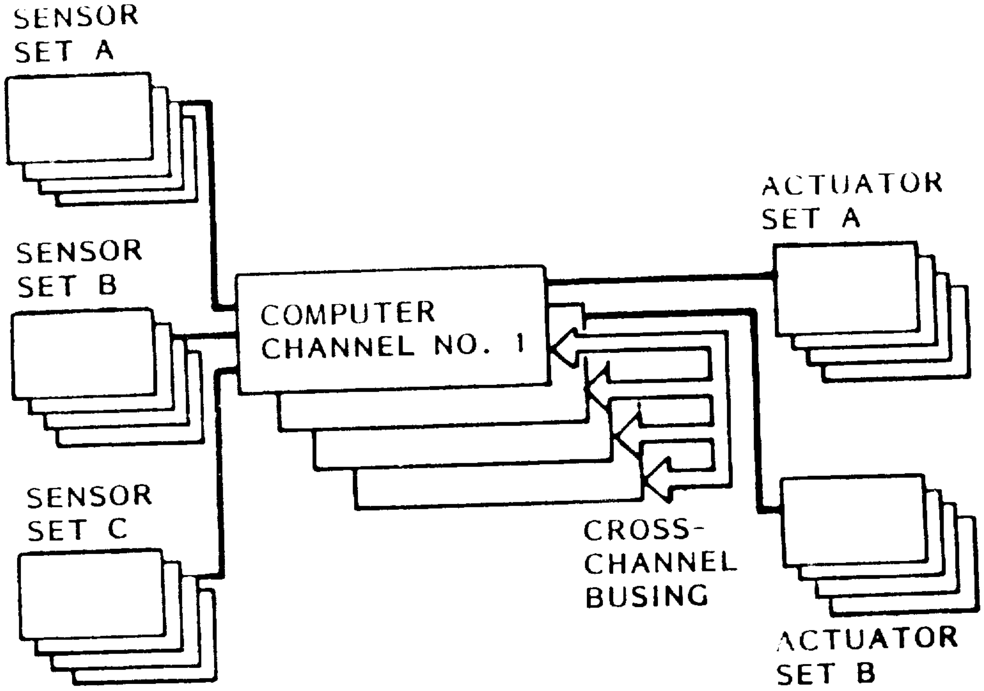


Figure E-1. Quadruplex DFCS Architecture

PRECEDING PAGE BLANK NOT FILMED

PAGE vi INTENTIONALLY BLANK

In all, the simulator modifications or enhancements were considerable. It was necessary to alter software in the PDP-11/60 to add relaxed static stability flight cases and new sensor signal outputs, and to add a real-time execution monitor. The new sensor signals in turn called for changes to the modular digital interface conversion unit (MDICU) program. Flight software states used by the monitor were obtained through the addition of instrumentation software in the PDP-11/04. Since the PDP-11/04 instrumentation response was insufficient for some purposes, one real-time execution monitor was programmed in one of the four flight computer channels to monitor the other three processors.

Finally, correspondence was established between design verification simulation of the fault-tolerant architecture design and real-time system results of the quadruplex DFCS. Essentially the same test cases and instrumentation parameters were used in both cases. Observability was superior in the case of the design verification simulation, for in some cases, implementation aspects increased the instrumentation task. The latter proved quite worthwhile in providing more confidence by confirming the executional correctness of low-level mechanization details.

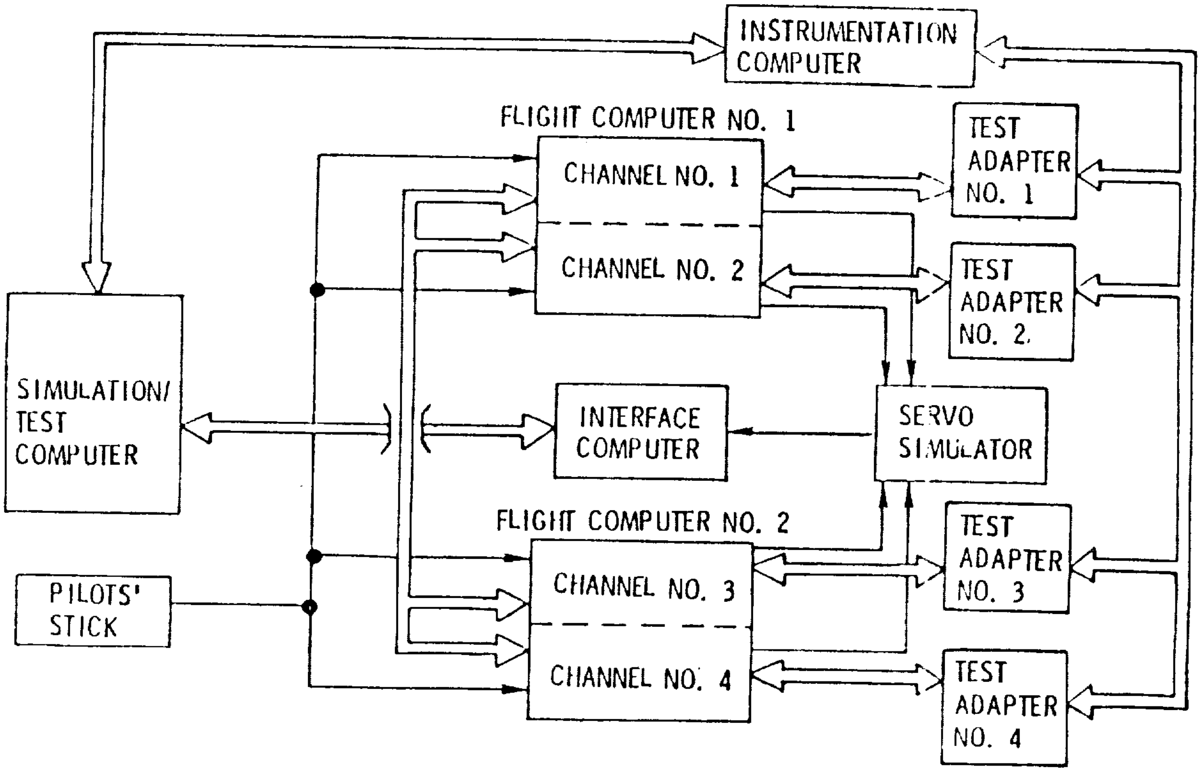


Figure E-2. RDFCS Facility Set-Up

1.0 INTRODUCTION

With the advent of full-time critical control functions, such as augmented fly-by-wire (AFBW) primary flight control systems, considerably more effort must be directed toward ensuring and confirming their safety than would be necessary for flight-phase critical functions such as autoland. As illustrated in this report, such effort involves the definition of fault-tolerant system architectures and the application of a suite of assurance tools to confirm system airworthiness. Since virtually all flight control systems are now implemented digitally, means must be employed to cope with greater inherent complexity than present in comparable analog mechanizations. The addition of fault tolerance mechanisms for achieving adequate system reliability, moreover, compounds this complexity problem.

At a system architecture level, the fault-tolerant system differences between analog and digital mechanization are only beginning to be very notable. But to attain adequate confidence levels in system airworthiness, lower levels of digital mechanization must be examined. This is where assurance methods and tools are essential. Much of the focus of much of this report, then, is directed toward the dependable attainment of the higher assurance levels stipulated in the FAA Advisory Circular 25.1309-1 (Reference 5). The approach taken here is the use of an integrated assurance methodology wherein the tools and methods are mutually reinforcing. Such an approach has previously been demonstrated at the system level (Reference 6).

Unfortunately, the requisite assurance methods, with very appreciable demands placed on them, have yet to be fully developed and cooperatively demonstrated. Overall, assurance levels of 10 exp -9 or less unreliability remain to be convincingly demonstrated in typical practice. This investigation offers some promising approaches to such concerns by way of an assurance driven methodology applied from initial design through real-time system simulation.

1.1 Assessment Rationale

Several dimensions of assurance method integration should be acknowledged in a comprehensive assessment process:

- o Reliability, failure effects, and functional performance assessment methods
- o Analysis, test, and inspection types of the above methods
- o Mutually supportive incorporation of all of the above in an assurance driven system development methodology.

Figure 1 depicts a central notion in the integrated assurance methodology used in the subject investigation. Basically, analysis is applied on a global scale to models or abstractions of the evolving DFCS configuration. Accordingly, analysis is the dominant assurance approach during the early stages of development, when only limited descriptions of the design are available. At that time inspections or walkthroughs are valuable as well, e.g., a review of the fault conditions applied in exercising the predicate/transition network simulation. Inspection comes into play, moreover, any time engineering judgment is exercised in determining the significance or validity of development process results.

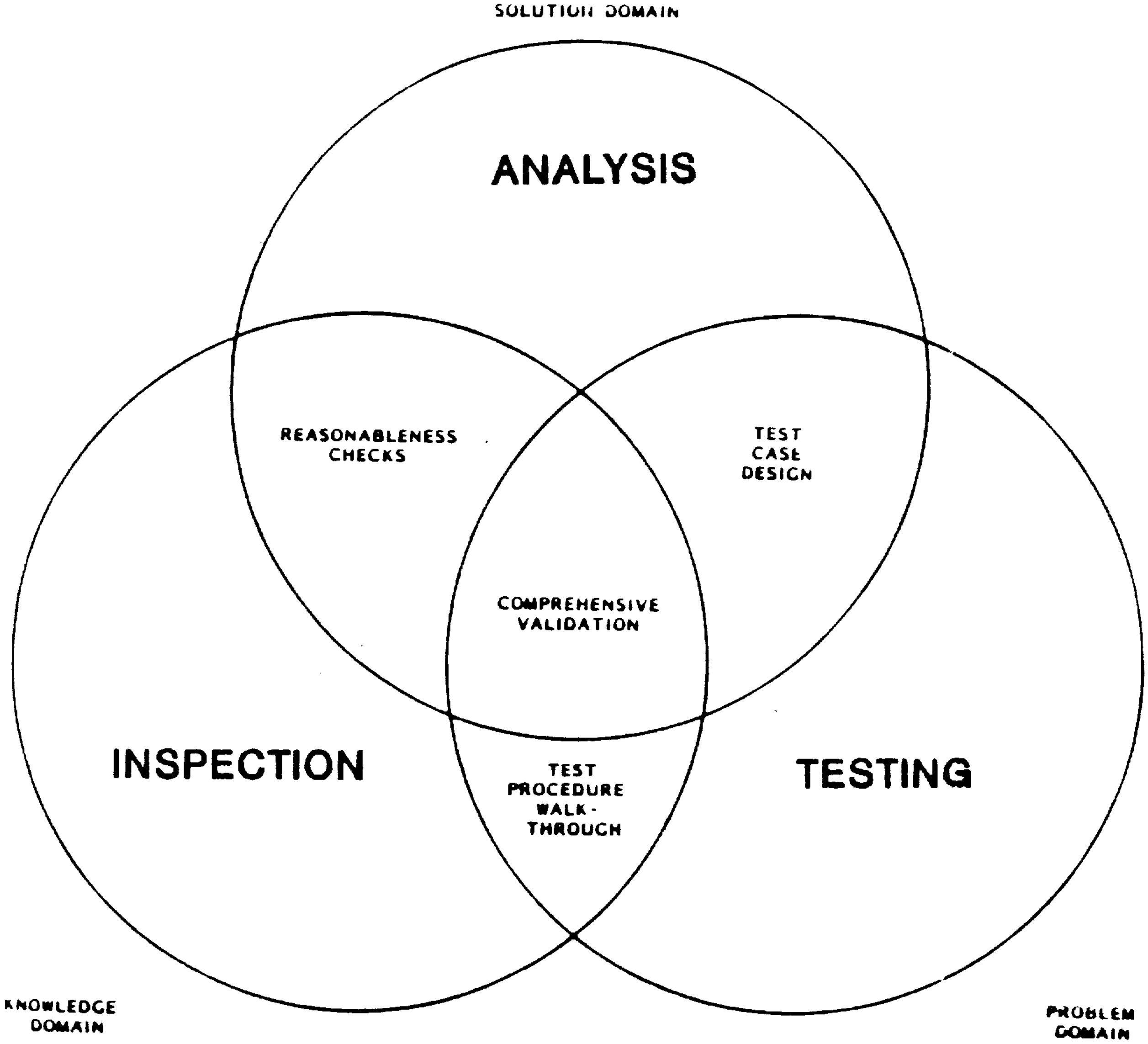


Figure 1. Complementarity of Assurance Methods

Since testing is usually considered to apply only on actual implementations, it takes place only after the test article has been mechanized. But the scope of practical test examination is necessarily limited, so only a small subset of possible test cases can be investigated. This situation gives rise to the major aspect of the complementarity of elements of an integrated methodology. Primarily, testing seeks to examine: the validity of selected analytical results; facets not amenable to analysis; assumptions underlying analyses; and operator-in-the-loop performance. Basically, testing is concrete, high fidelity, and readily convincing, but of it is lacking because of its inherently limited scope. Judicious test case selection is therefore vital in maximizing the assurances obtained through testing, but testing alone cannot provide adequate assurances.

Analysis, on the other hand, is abstract, idealized, and general, but very dependent upon proper formulation and interpretation. Analysis, moreover, is essential to effective testing. Various levels of analysis are involved in maximizing the conclusiveness of testing, as through the coincident, multilevel testing approach shown in Figure 2. As each stage of development proceeds, associated analyses yield test case definitions that can affirm that the ultimate implementation has remained in accord with prior design decisions. The fact that the various test cases can be applied coincidentally indicates another dimension of integrated assurance, one that can greatly extend validation process confidence and productivity.

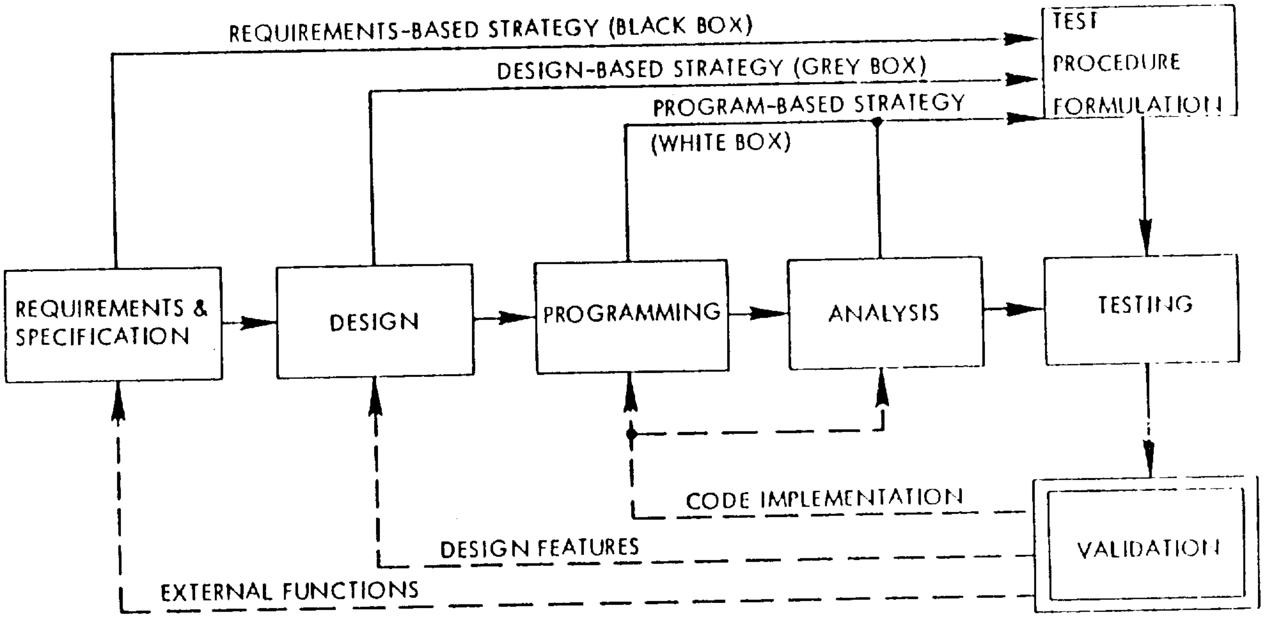


Figure 2. Basis of Multilevel Testing

1.2 Relevance to Other Tasks

This task is closely related to the N-version software fault tolerance task performed under the same contract (Reference 7). Basically, the same quadruplex DFCS design was used in both cases. Here, the executive software was implemented in AED, and system redundancy management issues were examined in a real-time system simulator. The N-version investigation focused mainly on Ada implemented applications software and its fault tolerance; a non-realtime test harness that supplanted the executive software was used so that the four channels of applications software could be run logically in parallel. Had the quadruplex DFCS been implemented in Ada, it would have been relatively easy to add the N-version software to it. As it turned out, the compatibility of the two task products proved useful for tutorial purposes.

Additionally, the task on analytical sensor redundancy was quite closely related to the subject one (Reference 8). Specifically, the analytical redundancy algorithms were added to the quadruplex DFCS software for the pitch stability augmentation sensors. In summary, these three tasks addressed computer hardware faults, sensor hardware faults, and DFCS applications software faults, all within the context and particular design constraints of the quadruplex DFCS architecture in this report.

2.0 BACKGROUND

There is an appreciable difference between the airworthiness assurances demanded of a full-time critical function, which is always required for flight, and a flight-phase critical function, which may experience very limited use. In particular, a primary flight control system is absolutely necessary at all times for safe flight, whereas a Category IIIa autoland is seldom used under actual Category IIIa weather minimums. While safety of a critical function must be assured in both cases, the risk in the former instance is far greater because of exposure time and severe limitations on alternatives. When such functions are mechanized digitally, there is presently concern over the capacity to ensure DFCS airworthiness. As a consequence, this investigation has sought to demonstrate the associated technology and its application in a representative DFCS development problem for a pitch-axis AFBW.

2.1 Terminology

In this report, the term ASSURANCE TECHNOLOGY is used in a very general sense to apply to all methods and activities for achieving or confirming the acceptability of a system. Primary emphasis, moreover, applies to the property of AIRWORTHINESS, or the assured safety of the system/vehicle. Those system functions whose proper operation is in general necessary for the safe operation of the aircraft are designated as CRITICAL per FAA AC 25.1329-01 (Reference 1). Functions that can noticeably deteriorate, but not preclude, the capacity for safe operation of the aircraft, are called ESSENTIAL.

CERTIFICATION refers to the formal process whereby the FAA authorizes deployment of an aircraft or system in response to evidence substantiating that each is indeed airworthy. Desirably, this process is supported with methods and tools over the development cycle that facilitate or ensure the conclusiveness of the evidence. The DFCS development cycle culminates with system VALIDATION, or confirmation that user requirements have been satisfied. Since these necessarily encompass system airworthiness, major emphasis in this report is placed on validating requisite fault survivability.

Of course the assurance process is a cumulative one that endeavors to attain increasingly convincing evidence of aircraft/system acceptability. Prior to product validation then, there are a series of VERIFICATION steps wherein compliance with various levels of specification is demonstrated. Of particular note here is system design verification, because as the first critical step in assuring the emerging system, it establishes the caliber and credibility of the overall assurance process.

In the earlier stages of system development, for example, considerable reliance is placed on analysis to confirm acceptability on a global or general basis. As the system is implemented, greater reliance is placed on testing particular aspects of the product. But properly, such testing derives from and reinforces the prior analyses (e.g., see Reference 6). In summary, note that verification ensures that a system is being "constructed right," or per specification, and that validation confirms that the "right system," or what the user wants, is being constructed (Reference 9).

2.2 FAA Regulatory Needs

Prior to the introduction of DFCSs, technical leadership within the FAA recognized and addressed the attendant challenges in the mid-1970s (Reference 10), as evident by the number of DFCSs certificated to date. These have limited function criticality, however, and with the current prospects for substantial or full-time reliance on critical DFCS functions, the FAA has again updated its digital technology agenda (see Reference 11). Basically, the FAA perceives a need within its regulatory staff to become familiarized with the nature and assurance challenges of the emerging critical systems.

Particular attention has been directed toward stability augmentation systems (SASs) and fly-by-wire (FBW) systems as soon to appear on commercial transports. Aside from function criticality per se, the issue of flying qualities under faults must be confronted regarding minimum safety over admissible flight profiles. This opens new areas of concern for the FAA, ones that ought to be illustrated on a demonstrator program of a tutorial nature.

2. 3 General Problem

From the reliability standpoint, the analytical resolution demanded for critical functions is a problem in itself for digital flight systems, and the cumulative degradation of flying qualities under multiple faults must be factored into the analysis. Other related aspects of the problem include dissimilar redundancy, back-up systems or instruments, fault transients recovery, and pilot workload limits. Interdisciplinary in nature, the overall problem is new as far as the certification of civil transports is concerned, so there is a serious and immediate need to formulate a unified assessment approach.

2.4 System Simulator Role

System simulation with system components, and possibly a pilot, in the loop is a vital and central part of flight controls practice, both before and after the advent of digital flight controls. To a lesser extent, developmental flight simulators have been used as well, especially for stability or control augmented military vehicles. Some of the work undertaken on this project is intended to develop and demonstrate better utilization of system simulation, but owing to the lack of an acceptable pilot interface in the RDFCS facility, a full simulator assessment of the quadruplex DFCS could not be demonstrated. Hence, this report only undertakes to describe the contributions of the pilot-in-the-loop role of flight simulators.

In the case of both system and flight simulators, the fidelity provided by system elements in the real-time loop is especially important in the case of digital systems because the inherent extra phase lag due to data holds and transport lags tends to seriously degrade system performance. Add to this the non-minimum phase characteristics of human pilots, and there is a suitable basis for delineating of minimum safe flying qualities. Since the associated DFCS control law analysis may not acknowledge these effects, the simulators become especially vital for digital systems.

Since the newer DFCSs are accompanied by electronic displays and other cockpit innovations, the pilot-in-the-loop performance has new aspects to be assessed. For example, the reversion to back-up electromechanical attitude displays under degraded flying qualities or stressful operational conditions may well alter what constitutes minimum safe flying qualities. Ultimately, such questions may not be difficult to resolve, but they do remain to be addressed coherently.

2.5 Assurance Technology

A rather broad definition of assurance technology was given in Section 2.2. Here it suffices to note that assurance features can be designed into a DFCS in a way that greatly facilitates the conduct and conclusiveness of the assurance process; this is an old theme that is particularly apt for fault-tolerant systems (see References 12 and 13). Since the testing reported here is for system validation, especially, under simulated faults, the overall assessment theme is to illustrate how accountability is propagated into the validation stage and what this stage contributes to the process.

2.6 Project Approach

In the interest of focusing the resources available for this task, an existing quadruplex pitch axis DFCS installed in the RDFCS facility was used as a test article. Developed under a Lockheed-funded project, this configuration resulted primarily from flight software changes to the basic single fail-operational, dual-dual RDFCS to obtain a double fail-operational systems. Although some constraints were imposed, the resulting system is quite representative of conventional quadruplex systems. Furthermore, since it was originally developed to demonstrate a rigorous system/software design methodology (see Reference 14), it was especially well-suited for use in an overall assurance assessment example.

3.0 OBJECTIVES

This quadruplex DFCS assessment investigation was intended to illustrate a critical DFCS assurance process, with focus on airworthiness certification topics. The investigation was largely motivated by prospects for certification technology tutorials. The methods illustrated were to be technically sound, pragmatically constructive, and where possible, innovative relative to the state of practice. The methods employed, however, are not necessarily held to be the best or the only effective ones for critical DFCS use.

3.1 Goal

This work is intended to provide a plausible and representative example of how to assure the airworthiness of full-time flight-critical digital flight systems in the near future. although addressed at a scaled-down level, the double fail-operational quadruplex DFCS problem is representative of near-term certification challenges, as for SAS or AFBW systems. the emphasis, however, is on the integrated use of assurance methods, automated tools for system simulator testing, and analytical test case definition. hopefully, the results obtained will foster a meaningful advance in the state of certification practice, with particular benefits accruing to the productivity and conclusiveness of validation testing.

3.2 Specific Objectives

The overall objective of this project has been the establishment of airworthiness technology readiness for fault-tolerant system architectures for full-time critical functions. Aside from the development and demonstration of the relevant technology, attainment of this objective is crucially dependent on the dissemination of results, both in report form and in tutorial workshops. Realization of the overall objective, moreover, has been predicated on the following elemental objectives:

- o Quantitative analytical comparison of basic DFCS redundancy levels relative to system reliability
- o Quantitative analytical comparison of alternative quadruplex configurations relative to system reliability
- o Critique of critical AFBW mechanization in terms of fault survivability
- o System simulator demonstration of basic quadruplex DFCS fault survivability, in part using automated test methods.

PRECEDING PAGE BLANK NOT FILMED

PAGE 8 INTENTIONALLY BLANK

3.3 Scope

As noted in Section 2.4, this investigation of augmented flying qualities was conducted without an acceptable pilot interface. This resulted from RDFCS facility limitations and funding level realities. Also, the quadruplex DFCS was implemented almost solely through software modifications, so while functionally representative, certain aspects usually implemented in hardware were rendered in software. Only pitch axis flight control functions were incorporated in the DFCS, but these served to illustrate the newer type critical functions. In all, a balance was sought between available resources and realizable results, and where project economies were necessary, this report has attempted to address and elucidate the associated technical issues and consequences.

3.4 Expectations

Since this effort has focused primarily on system architectures, there remains a need to explore complementary aspects of minimum safe stability augmentation functions using pilot-in-the-loop flight simulation. The need encompasses both methods and criteria, as well as a means to incorporate the associated results into the overall assurance process. Of course flight simulation entails a high fidelity flight station with electronic displays and appropriate controllers. It is therefore projected that this type undertaking will soon be pursued by a multidisciplinary team in the context of commercial transport applications.

In the farther term, as fault-tolerant architectures become more sophisticated, the role of the system simulator is expected to be diminished by a more general and powerful system development facility known as a rapid prototyping environment. Such a facility places greater emphasis on assurance activities on the front-end of the development cycle, and subsequently propagated accountability. It is therefore projected that such facilities and their enabling methods and tools will evolve over the next decade to become standard type installations within the airframe business.

4.0 TASK RESULTS

With existing RDFCS constraints in mind, a representative double fail-operational DFCS architecture was defined from among a set of related candidates. The redundancy management coordination of this design was verified through predicate/transition network simulation, and the high-level software design was then represented in nested control graphs. The actual flight code was rendered in AED, beginning with an austere real-time executive, and loaded into the Collins CAPS-6 flight computers at NASA Ames. Cross-channel coordination was effected through respective channel control states broadcasted over corresponding serial digital buses. The quadruplex DFCS was then tested under various simulated fault and anomaly conditions using real-time software execution monitors to resolve low-level system management events.

RSS flight cases were installed in the PDP-11/60 flight simulation at NASA Ames, and validated using previously defined airplane root solutions and time history check cases. New AFBW sensor outputs were ported from the PDP-11/60, through the MDICU, to the flight computers. Also, new mode and fault logic signals were assigned on the logic discrete switch panel on the simulator pallet. Lastly, the pitch-axis manual control stick inputs were introduced into the flight computers from a hand controller.

Hence, it was possible to assess flying qualities degradation through real-time closed-loop system simulation, with or without an operator in the loop. Unfortunately, the manual flying qualities assessment was hampered by the poor quality of pilot interface available, but the characteristics of the free or unaugmented airplane yielded such severe or noticeable degradation under some sensor faults that the interface was actually of some use.

4.1 System Definitions

System definition evolved per the development products noted in Figure 3, which depicts mechanization increments along the downward path on the left, and assurance milestones along the upward path on the right. Further detail on the development activities are presented in Table 1 within the framework of the typical stages of development. Certain of the aforementioned development products are illustrated later in this report. **The intent here is to exemplify key steps, corresponding to progressive system definitions, that should lead to a certifiable DFCS.**

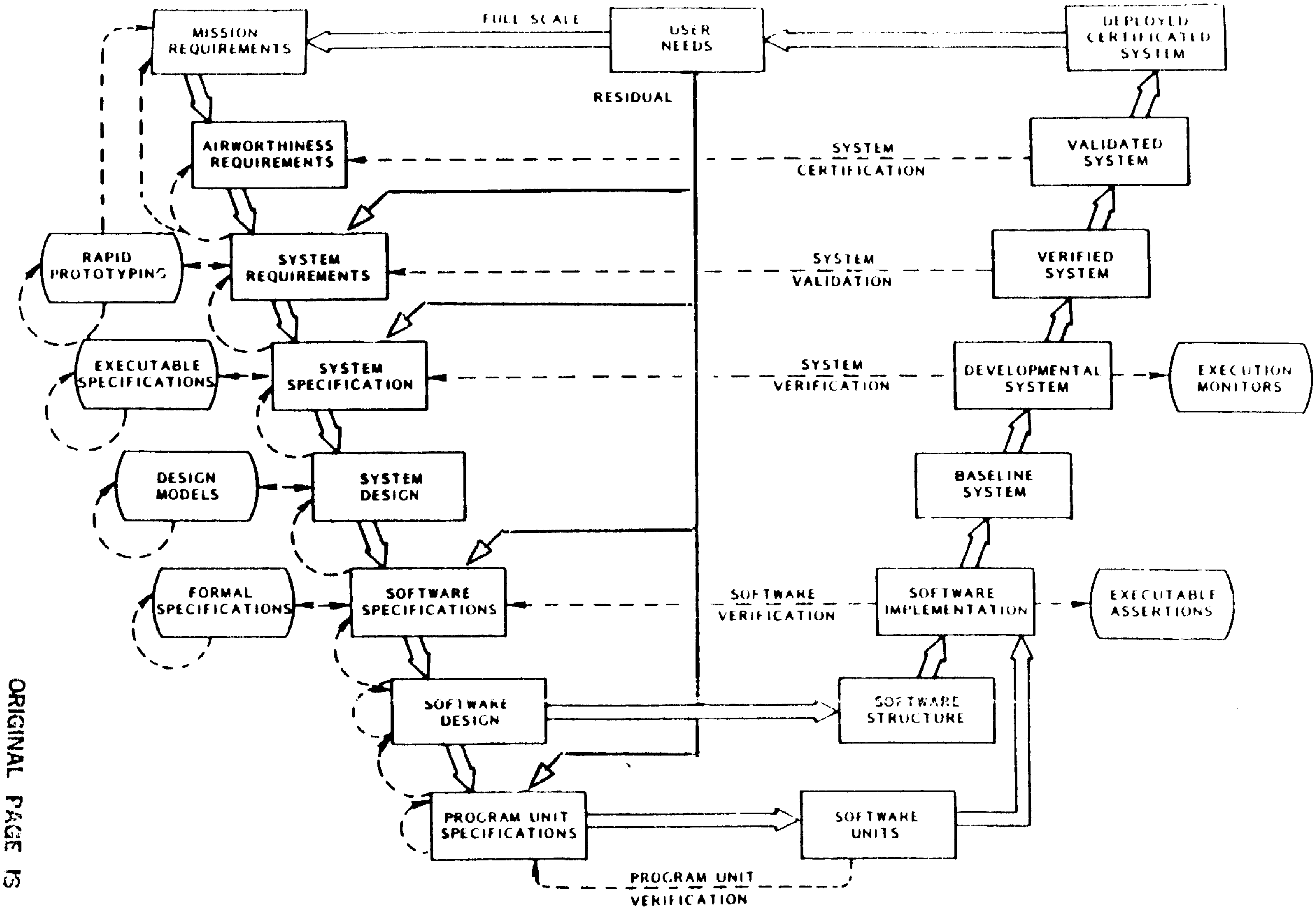


Figure 3. Digital Flight System Life Cycle

ORIGINAL PAGE IS OF POOR QUALITY

Table 1. Development Phase Activities and Products

PHASE	PURPOSE	INPUT	PROCESS		DEVELOPMENT PRODUCT	SYSTEM STATUS
			MECHANIZATION	ASSURANCE		
CONCEPTUAL	"PROVIDE FOR USER NEEDS"	MISSION REQUIREMENTS	FORMULATE SYSTEM REQUIREMENTS EXPLORE DESIGN SOLUTIONS	VALIDATE REQUIREMENTS ANALYZE DESIGN SOLUTIONS	SYSTEM REQUIREMENTS SYSTEM CONCEPTS	FUNCTIONAL ARCHITECTURE "FEASIBLE DESIGN SOLUTION"
DEFINITION	"DESIGN FOR USER NEEDS"	AIRWORTHINESS REQUIREMENTS, SYSTEM REQUIREMENTS, SYSTEM CONCEPTS	FORMULATE SYSTEM SPECIFICATION REFINE SYSTEM CONCEPTS	VERIFY SPECIFICATIONS VALIDATE SYSTEMS CONCEPTS	SYSTEM SPECIFICATION CONCEPTUAL SYSTEM DESIGN	VALIDATED CONCEPTUAL DESIGN "ACCEPTABLE DESIGN SOLUTION"
ANALYSIS	"DESIGN SYSTEM TO SPECIFICATION"	SYSTEM SPECIFICATION	DESIGN SYSTEM STRUCTURE, DESIGN CONTROL LAWS	VERIFY SOFTWARE SPECIFICATION, VERIFY SYSTEM STRUCTURE, VERIFY CONTROL LAWS	SYSTEM DESIGN SOFTWARE SPECIFICATIONS, HARDWARE SPECIFICATIONS	VERIFIED SYSTEM STRUCTURE "SUPERIOR DESIGN SOLUTION"
DESIGN	"DESIGN SOFTWARE TO SPECIFICATION"	SOFTWARE SPECIFICATION	DESIGN SOFTWARE STRUCTURE, DEFINE SOFTWARE COMPONENTS	VERIFY SOFTWARE DESIGN, VERIFY UNIT SPECIFICATIONS	SOFTWARE DESIGN PROGRAM UNIT SPECIFICATIONS	VERIFIED BASELINE DESIGN "COMPREHENSIVE DESIGN DEFINITION"
CODING AND CHECKOUT	"IMPLEMENT SOFTWARE TO SPECIFICATIONS"	UNIT SPECIFICATIONS	IMPLEMENT PROGRAM UNITS	CHECK/DE-BUG UNITS, VERIFY PROGRAM UNITS	SOFTWARE IMPLEMENTATION	BASELINE SYSTEM CONFIGURATION "COMPREHENSIVE SYSTEM DEFINITION"
INTEGRATION	"CONSTRUCT SYSTEM WITH HARDWARE/SOFTWARE COMPONENTS"	VERIFIED STRUCTURE AND COMPONENTS	ASSEMBLE/DEVELOP SYSTEM	IDENTIFY/RECTIFY INCONSISTENCIES	SYSTEM IMPLEMENTATION	DEVELOPMENTAL SYSTEM
DEVELOPMENT TEST	"TEST TO SPECIFICATION REQUIREMENTS"	SYSTEM SPECIFICATIONS	DEVELOP SYSTEM OPTIMIZE PERFORMANCE	IDENTIFY/RECTIFY DEFICIENCIES VERIFY PERFORMANCE	VERIFIED IMPLEMENTATION	VERIFIED SYSTEM
VALIDATION	"TEST FOR REQUIREMENTS COMPLIANCE"	SYSTEM REQUIREMENTS	MODIFY SYSTEM IF NECESSARY	CONFIRM ACCEPTABILITY	PROVISIONAL CONFIGURATION	VALIDATED SYSTEM
CERTIFICATION	"DEMONSTRATE AIRWORTHINESS COMPLIANCE"	AIRWORTHINESS REQUIREMENTS CERTIFICATION PLAN	MODIFY SYSTEM IF NECESSARY	CONFIRM AIRWORTHINESS	PRODUCTION CONFIGURATION	CERTIFICATED SYSTEM

ORIGINAL PAGE IS
OF POOR QUALITY

Broadly, design definitions usually address either system function(s) or architecture. The former centers on control laws, and the latter expands into a detailing of system/software structure. With regard to the structure of digital mechanizations, it is particularly vital to explicitly describe the organization of the flight software. These three aspects of system definition are discussed further in the following subsections prior to presenting application examples. Because of its centrality to reliable/fault-tolerant DFCSs, particular emphasis is placed on system/software architecture. Hence, the design tasks noted in Figure 4 have been illustrated through a sequence of example development stages.

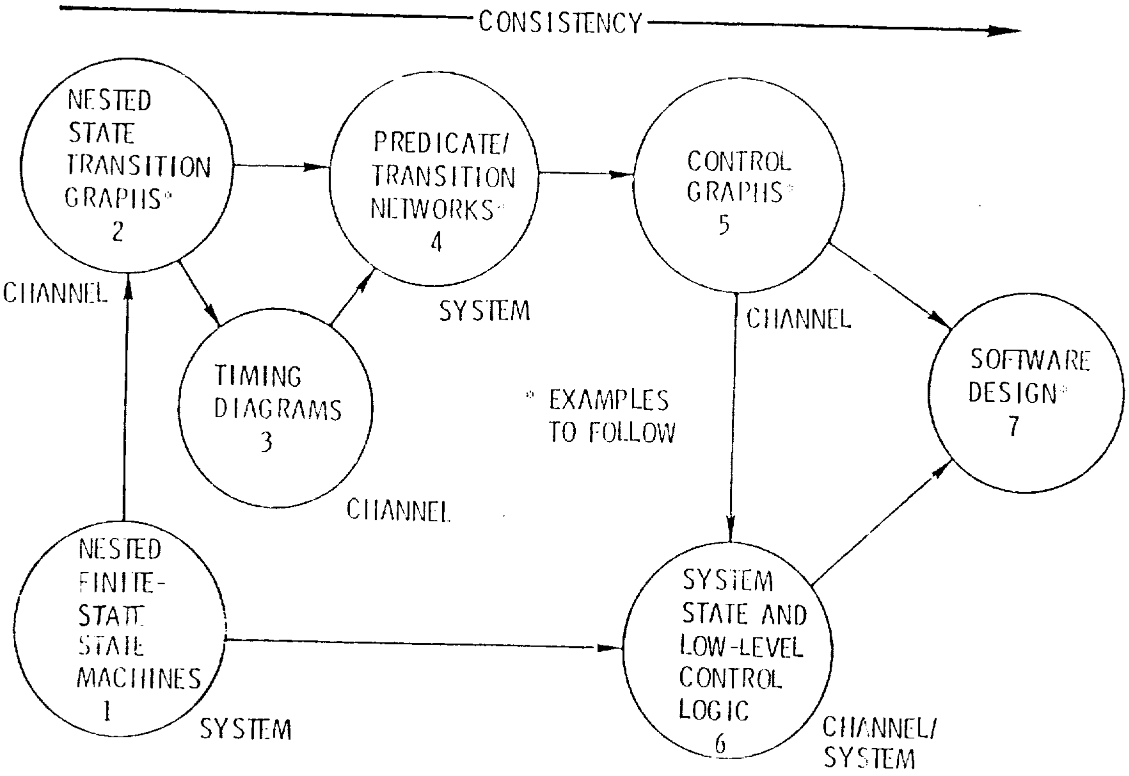

CONSISTENCY
NESTED STATE TRANSITION GRAPHS* 2
PREDICATE/ TRANSITION NETWORKS* 4
CONTROL GRAPHS* 5
CHANNEL
SYSTEM
CHANNEL
TIMING DIAGRAMS 3
* EXAMPLES TO FOLLOW
SOFTWARE DESIGN* 7
CHANNEL
NESTED FINITE-STATE STATE MACHINES 1
SYSTEM
SYSTEM STATE AND LOW-LEVEL CONTROL LOGIC 6
CHANNEL/ SYSTEM

Figure 4. Architecture Design Tasks

4.1.1 Quadruplex System Description

Most DFCSs to date have essentially been digitized versions of previous analog mechanizations. These DFCSs have been characterized by: conventional system architectures based upon parallel replication; frequency domain derived control laws mapped into difference equations; and multirate foreground executive programs to periodically call appropriate applications software for selected control functions. In general, systems have been dedicated strictly to control functions, thereby avoiding potentially debilitating software anomalies resulting from interference by other software functions.

Parallel channels, each with virtually the same software, have only recently yielded to dissimilar processors or software to reduce the possibilities for coincident or generic design faults. Similarly, floating-point arithmetic processors are beginning to replace fixed-point processors; this change obviates the need for scaling variables, another carry-over from analog computer practice. Higher-order programming languages are becoming prevalent for quality and cost reasons, and Ada will likely soon dominate software on commercial aircraft. Multiplex (MUX) data buses are now predominant over point-to-point buses for broader and more general usage of resources.

4.1.1.1 Pitch Augmentation Function

To pose a full-time criticality problem, a relaxed static stability transport airplane was defined with a negative stability margin and a fly-by-wire primary flight control system. The associated control law depicted in Figure 5 is employed in each of four computational channels, as indicated by the quad input voters. Pilot commands are applied through control stick inputs; short-period damping is provided by actual, not derived, pitch rate; and angle-of-attack is used to control pitch-axis divergence associated with RSS. Special signal processing includes control stick deadbands and pilot/copilot stick blending, left/right angle-of-attack averaging, and command limiting.

Because control surface effectiveness varies over the flight profile, certain DFCS gains must do likewise. This facet of design is referred to as gain scheduling, and true airspeed is often used to schedule control law parameters. In this investigation, six point simulation flight cases were employed, so Table 2 presents their associated sensor feedback gains. Note that the gains are generally less at higher speeds because control surface effectiveness tends to increase with speed.

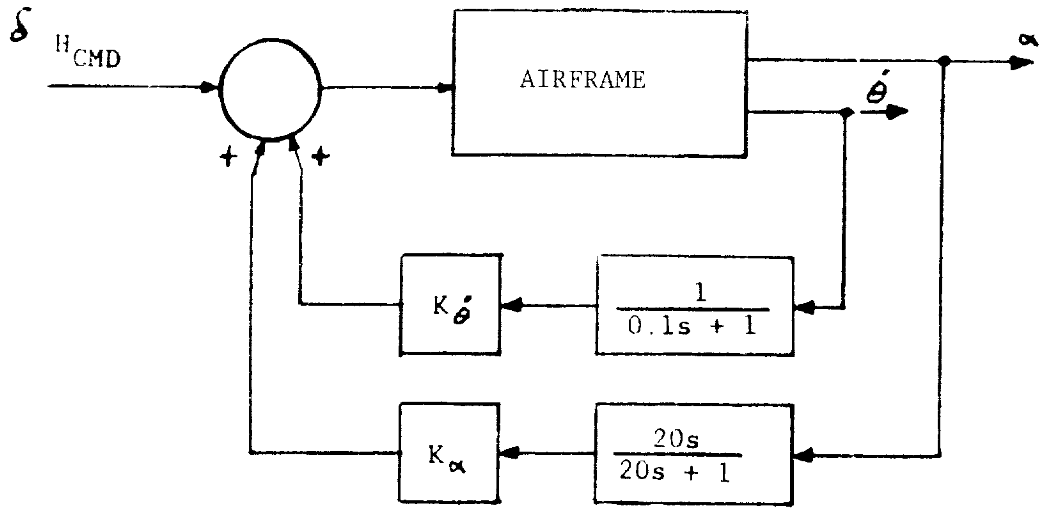


Figure 5. Augmented Fly-by-Wire Control Laws

Table 2. Stability Augmentation Gain Scheduling

FLIGHT CASE	TRUE AIRSPEED (fps)	K_{α} (deg./deg.)	$K_{\dot{\theta}}$ (deg /deg. per sec.)
A1RSS	283.7	1.00	0.70
C13RSS	910.7	0.73	0.10
C15RSS	442.2	1.00	0.40
D2RSS	487.1	1.00	0.33
E3RSS	265.7	1.00	0.80
F6RSS	224.1	1.00	0.50

4.1.1.2 Basic System Architecture

The first DFCS architectures introduced were rather conventional in that they largely reflected prior analog configurations, at least at the block diagram level. These employed parallel, replicated channels wherein sensors fan-in to processors, which in turn fan-out to effectors or displays. Figure 6 presents an expansion of such an architecture for the pitch-axis AFBW function. Except for cross-channel communication, all signal paths are dedicated analog paths, and fan-in is minimized by having one sensor set applied directly to each computational channel. Each channel then broadcasts its received inputs to the other three. Other architectural options are described and critiqued later.

The top-level primary flight control system (AFBW) requirements are assumed to be MIL-F-9490D operational states (Reference 15). Because these have well defined meaning that encompasses both airplane flying qualities and system safety in terms of redundancy margins. Operational State 1 denotes normal system status and Level 1 or good flying qualities; State 2 admits some deterioration in safety margins and Level 2 or somewhat degraded flying qualities; State 3 indicates marginal safety margins and flying qualities (Level 3); and State 4 or worse designates unsafe componentry and/or flying qualities. Knowledge of flying qualities degradation under successive component failures enables the operational state logic to be viewed from strictly an architectural point of view. The system design task then focuses on channel-level logic definition to effectuate the system-level logic.

Figure 7 represents the top-level DFCS channel logic design, where the names within the circles correspond to particular states that associated logic variables can assume. These variables denote states or sub-states of the channel. The names appearing on the arcs of the state transition graphs designate independent logic events such as pilot mode selection, a timer interrupt, or a component failure. Such events in turn may effect changes in the channel states per se, which must reflect the state of the system. Note that the transition graphs are nested to reduce complexity; the lower-level graphs, moreover, capture sub-state information.

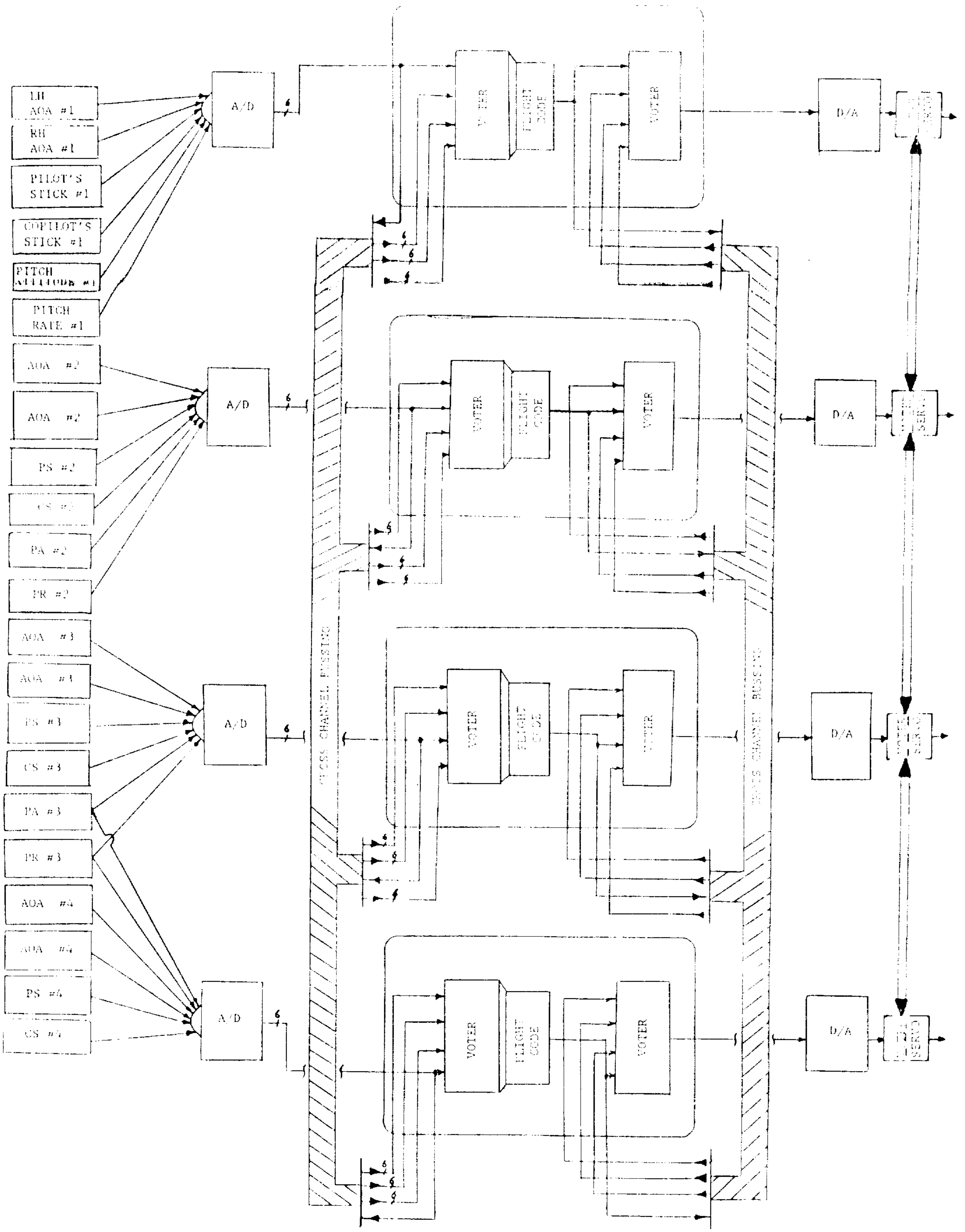


Figure 6. Baseline System Architecture

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

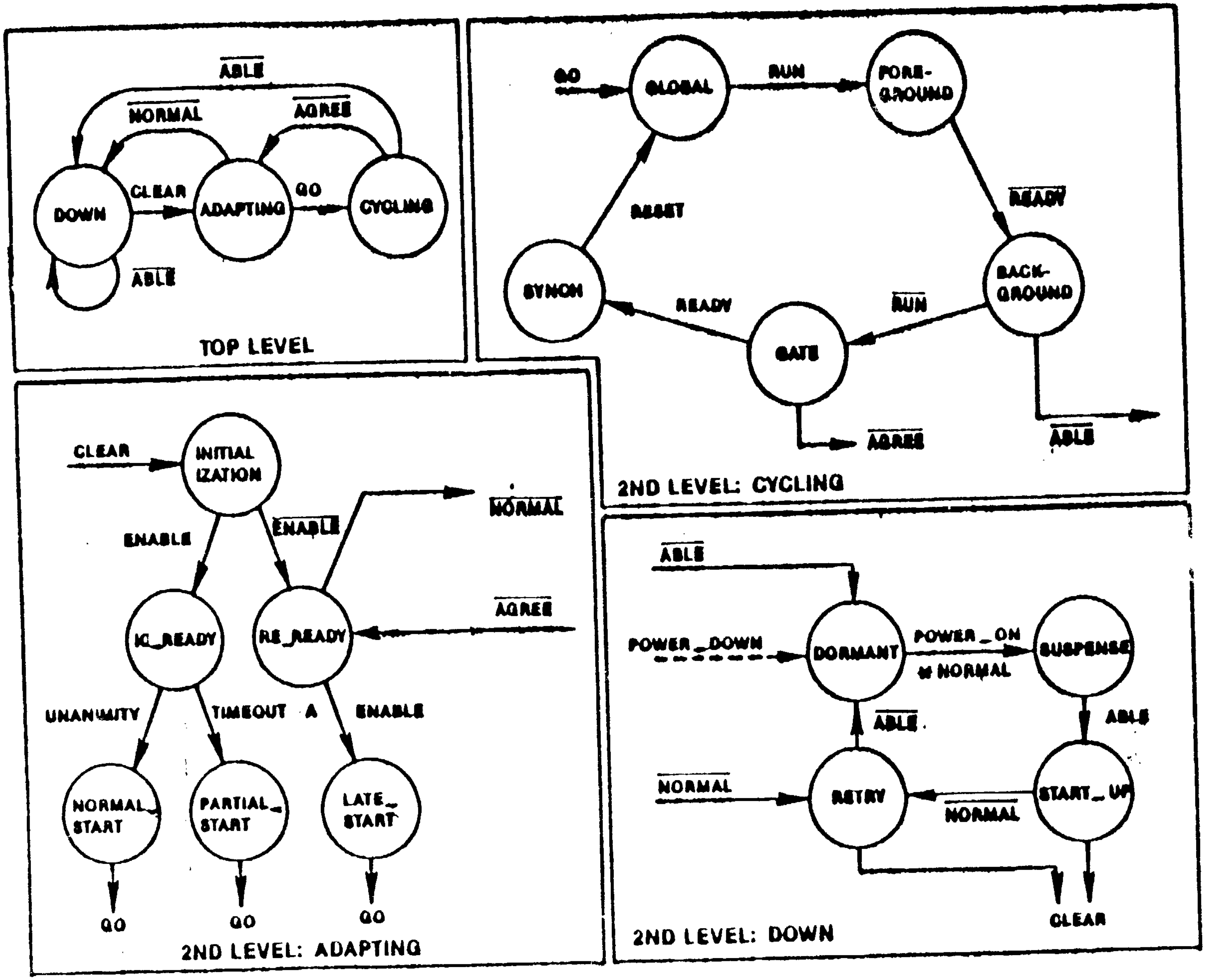


Figure 7. Channel Logic Transition Graphs

In the Gate Sub-state of the Cycling State, for example, each channel develops its view of system status based on control state information for each of the four channels. This type of consensus is one of the major determinants in calculating the operational state of the system. In all, the implementation of this design obviously necessitates an appreciable expansion of the logic in the flight code. First, however, it is necessary to examine the system level coordination logic meeting the timing constraints in Table 3.

System synchronization logic with a discrete time base imposed is captured in Figure 8, which is called a predicate/transition network (Reference 2). This view represents only one channel, but all four are the same for the subject architecture. Here the same logic nomenclature is retained where applicable, and some new logic variables are added. the logic names appear in the rounded boxes, which are called "places" in the network. At any given time, the collective values of all places constitute the state of the system modeled. To examine and verify the correctness of all possible state sequences, the network's operation must be simulated using a computer program. Properly accomplished, such a simulation, under both faulted and fault-free conditions, verifies the system logic design.

Table 3. Cycling State Loop Timing

	SUBSTATE	GLOBAL	FOREGROUND	BACKGROUND	GATE	SYNCH
	DURATION	0-1 MS	$1 - t_{fg}$	$t_{fg} - 48$	48-49	49-50
EVENT	RUN	FALSE	TRUE	TRUE	FALSE	FALSE
	READY	TRUE	TRUE	FALSE	FALSE	TRUE
	RESET	TRUE	FALSE	FALSE	FALSE	FALSE

- 50 MILLISECOND (MS) COMPUTATIONAL FRAME TIME
- $t_{fg} < 48$ MS
- TIMER INTERRUPT AT t = 48 MS => RUN
- RESET => t = 0

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

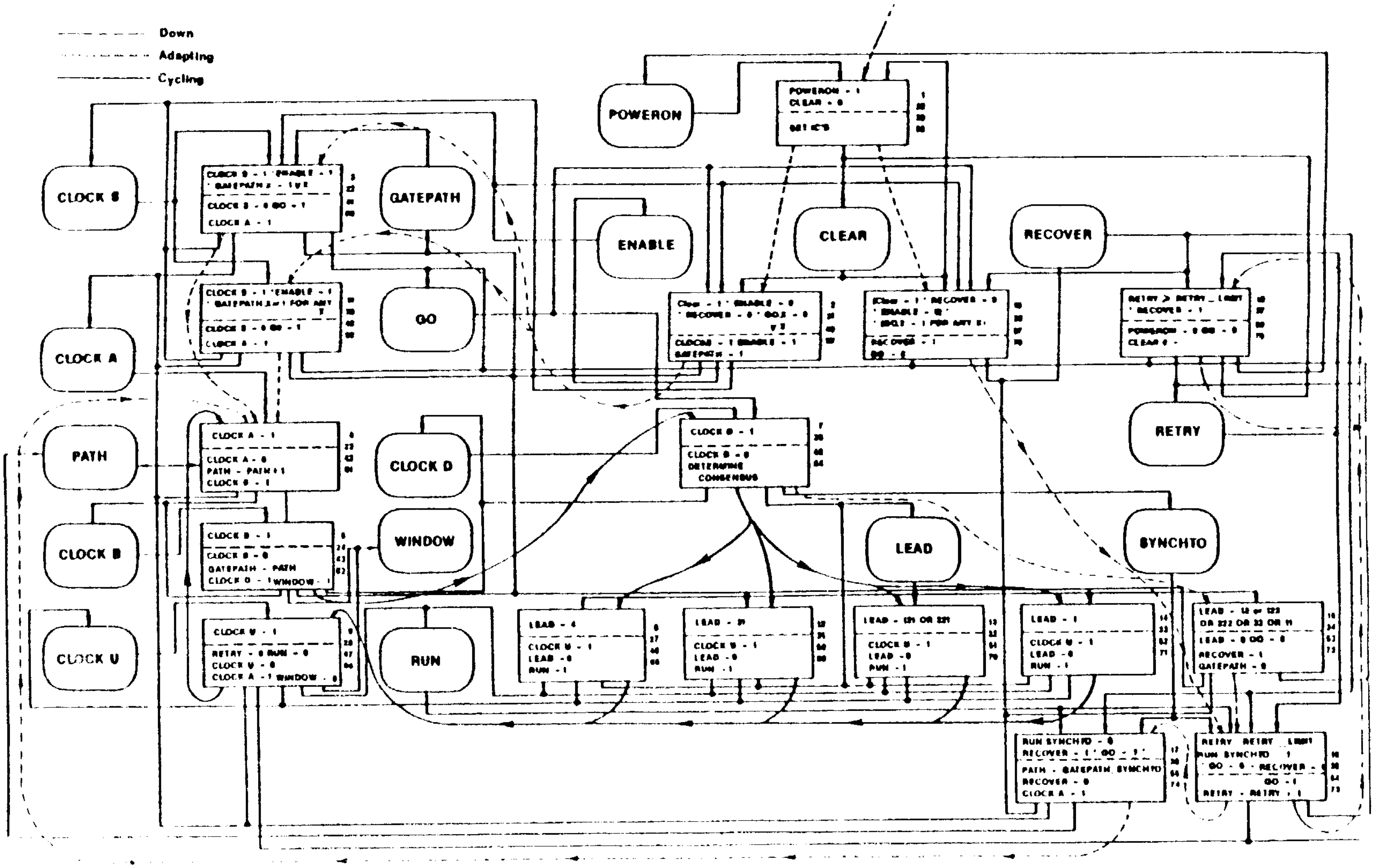


Figure 8. Synchronization Predicate/Transition Network

The simulation is based on the firing of "transitions," as denoted by rectangles, which yield new values for the logic variables stored in the places. The top half of each transition box describes a predicate whose satisfaction by system logic values enables it to be fired. Whenever a transition fires, the new logic values described in the bottom half of the rectangle are assigned. these values are then reflected in the appropriate places, and a new network state produces a new set of transition that can be fired.

This mechanism can be seen more clearly by noting the partial network in Figure 9. Here the network captures a design wherein the hardware initialization within a channel sets its POWER_ON to True and its CLEAR to False. This arms the top transition, whose firing corresponds to

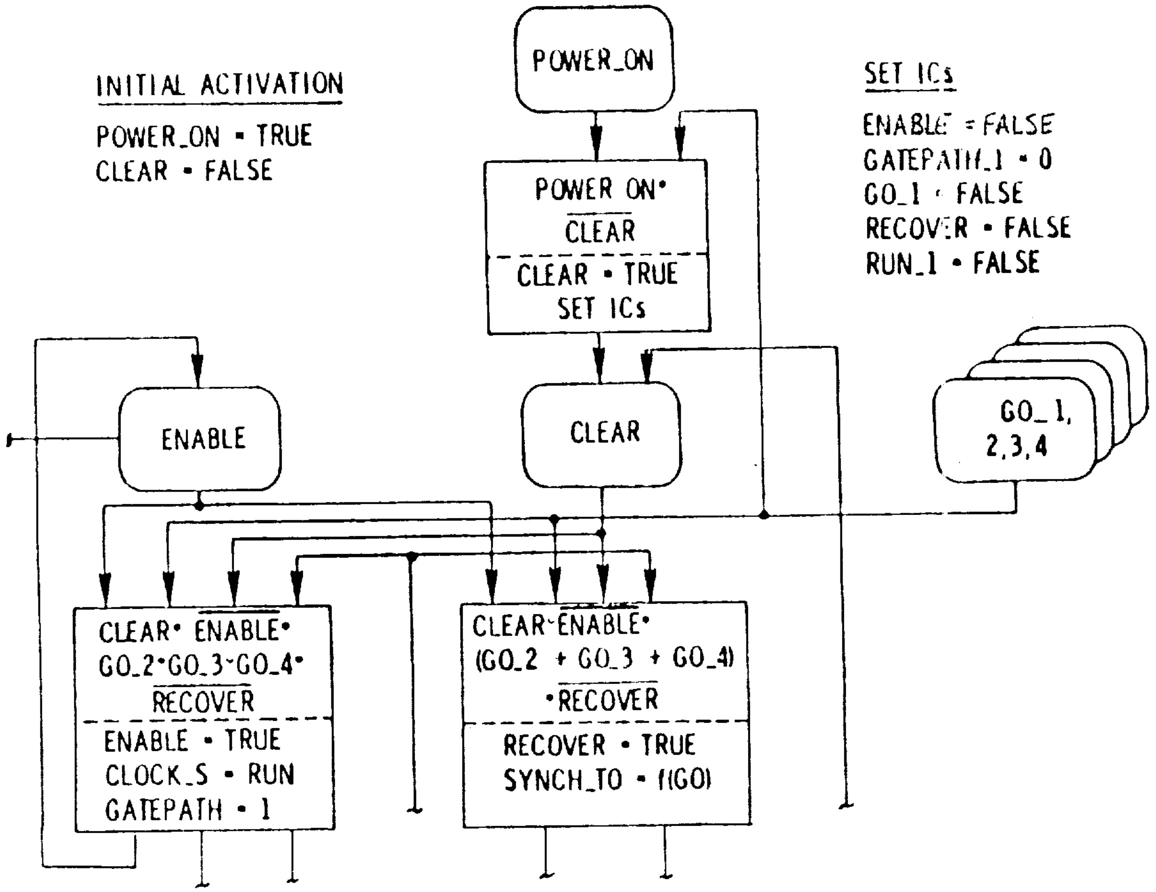


Figure 9. Predicate/Transition Network Detail

software initialization that sets CLEAR to True among other logic variable assignments shown. Referring to Figure 9, note that the event of CLEAR be set to True effects the top-level channel state transition from DOWN to ADAPTING. At this point, the channel tries to synchronize with the other channels, to enter the CYCLING state.

Network simulation should always yield acceptable system states, and should never terminate unless all channels are failed. Determining that this is the case is a matter of defining and obtaining correct logic operation. In the subject investigation, this was accomplished with one exception noted later. Figure 10 illustrates some typical discrete-event simulation results for four-way channel synchronization. The pulses at the top indicate instances wherein individual channels were forced out of synchronization, i.e., the corresponding logic variable RECOVER went True. The re-synchronization logic in the network model then satisfactorily restored synchronization, and the RECOVER was set to False as indicated by the end of the pulse in Figure 10.

Following synchronization design verification which captured time-based hardware/software interaction, the design emphasis shifted to the top-level software design with the constraints imposed by the existing RDFCS hardware.

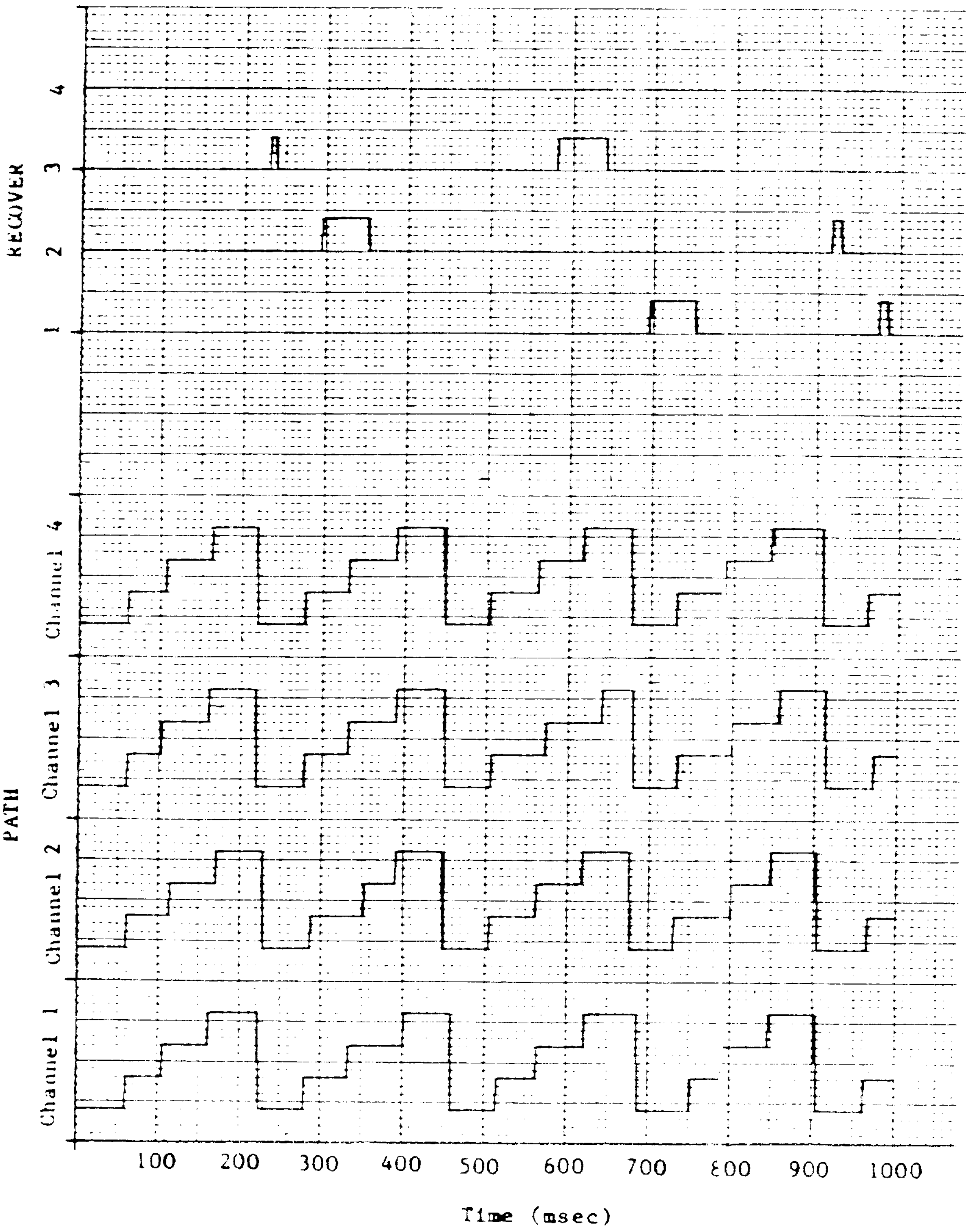


Figure 10. Predicate/Transition Network Simulation Output

ORIGINAL PAGE IS
OF POOR QUALITY

4.1.1.3 Executive Flight Software Organization

As with the basic Collins system, the flight software is hardware interrupt-driven at the given 60 Hz rate. Since the AFBW control laws were designed for 20 Hz operation, the executive software invokes the applications control functions every third interrupt. The executive program itself is rather austere, as appropriate for a dedicated system. Here the computational channels mutually coordinate themselves in a frame synchronous, double fail-operational manner.

To achieve this, the channel design described in the previous section must be mapped into a software design with objectives of: maintaining consistency in the system-to-software design transition; and minimizing the complexity of the software. Accordingly, the control graph in Figure 11 represents a mapping of the top-level transition graph in Figure 7 to a software control structure that preserves the design logic in a form exhibiting only moderate decision logic complexity per the cyclomatic number (see Reference 16). As by-products, the control graph yields test case input vectors and logic assertions that were later used in verifying the implemented code.

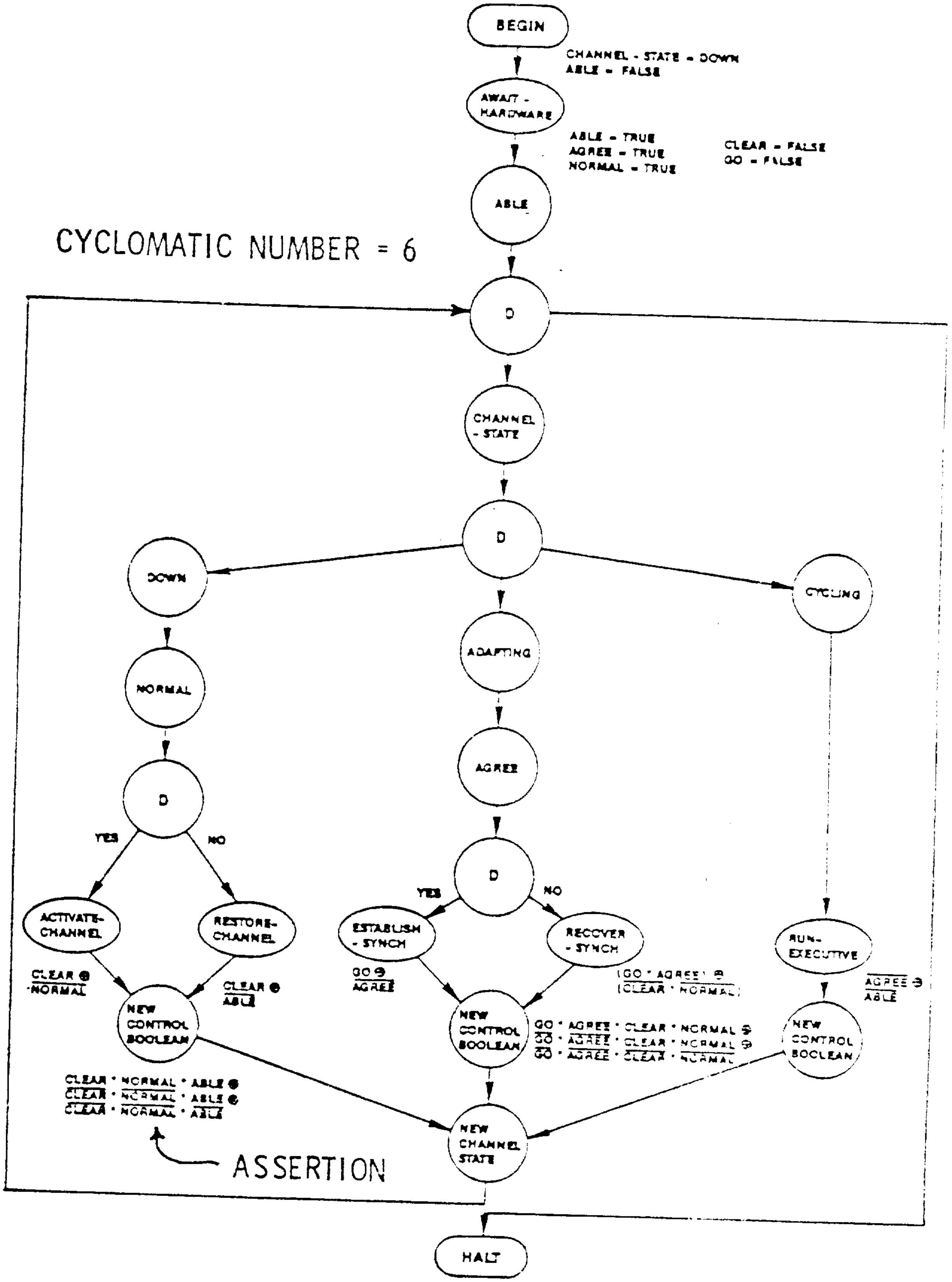

BEGIN
CHANNEL - STATE = DOWN
ABLE = FALSE
AWAIT - HARDWARE
ABLE = TRUE
AGREE = TRUE
NORMAL = TRUE
CLEAR = FALSE
GO = FALSE
ABLE
CYCLOMATIC NUMBER = 6
D
CHANNEL - STATE
D
DOWN
ADAPTING
CYCLING
NORMAL
AGREE
D
YES
NO
D
YES
NO
ACTIVATE-CHANNEL
RESTORE-CHANNEL
ESTABLISH - SYNCH
RECOVER - SYNCH
RUN-EXECUTIVE
$\text{CLEAR} \oplus \overline{\text{NORMAL}}$
$\text{CLEAR} \oplus \overline{\text{ABLE}}$
$\text{GO} \oplus \overline{\text{AGREE}}$
$(\text{GO} \cdot \text{AGREE}) \oplus (\overline{\text{CLEAR} \cdot \text{NORMAL}})$
$\overline{\text{AGREE}} \oplus \overline{\text{ABLE}}$
NEW CONTROL BOOLEAN
NEW CONTROL BOOLEAN
NEW CONTROL BOOLEAN
$\text{GO} \cdot \text{AGREE} \cdot \text{CLEAR} \cdot \text{NORMAL} \oplus$
$\overline{\text{GO}} \cdot \overline{\text{AGREE}} \cdot \text{CLEAR} \cdot \text{NORMAL} \oplus$
$\overline{\text{GO}} \cdot \overline{\text{AGREE}} \cdot \overline{\text{CLEAR}} \cdot \overline{\text{NORMAL}}$
$\text{CLEAR} \cdot \text{NORMAL} \cdot \text{ABLE} \oplus$
$\overline{\text{CLEAR}} \cdot \overline{\text{NORMAL}} \cdot \text{ABLE} \oplus$
$\overline{\text{CLEAR}} \cdot \overline{\text{NORMAL}} \cdot \overline{\text{ABLE}}$
ASSERTION
NEW CHANNEL STATE
HALT

Figure 11. Top-Level Software Control Graph

ORIGINAL PAGE IS
OF POOR QUALITY

4.2 Reliability Assessment

Reliability assessments were directed toward contrasting levels and types of redundancy relative to their impact on system reliability for critical functions. Table 4 delineates 12 different system architectures that were analyzed for critical system function failure on both five- and ten-hour missions. The system architecture in Figure 6 actually corresponds to Cases 1 and 2 in Table 4, depending upon whether inherent back-up capability is invokable. Because of the mode selection switch arrangement devised in the RDFCS laboratory, the back-up mode was manually selectable. Hence, Figure 12 represents the reliability model for Architecture Case 1, the one actually implemented. Note that the dependencies and logic embedded in this reliability model do not, however, apply to Case 2.

Cases 1 through 5 are all-up AFBW architectures, which are the primary concern here. Cases 1 through 3 meet the critical function reliability requirements of Reference 1, but the analyses only take into account hardware fault contributions to unreliability. Still, the data are instructive in several ways. Basically, the back-up pitch hold mode offers surprisingly little to survivability, partly because it would be needed so infrequently, and at a time when it too might well be unavailable due to the loss of common components with the basic pitch SAS.

Cases 4 and 5 are inadequate because of reduced redundancy levels. Cases 5 through 7 isolate the reliability properties of straight FBW architectures. A contrast of Cases 2 and 6, for example, discloses that the critical pitch SAS function increases the probability of failure by about only a third for straight FBW. the straight SAS function is isolated in Cases 9 through 12, and it is noteworthy that triplex AOAs and computers are inadequate per Case 12. Triplex servos, with quadruplex sensors are, however, satisfactory.

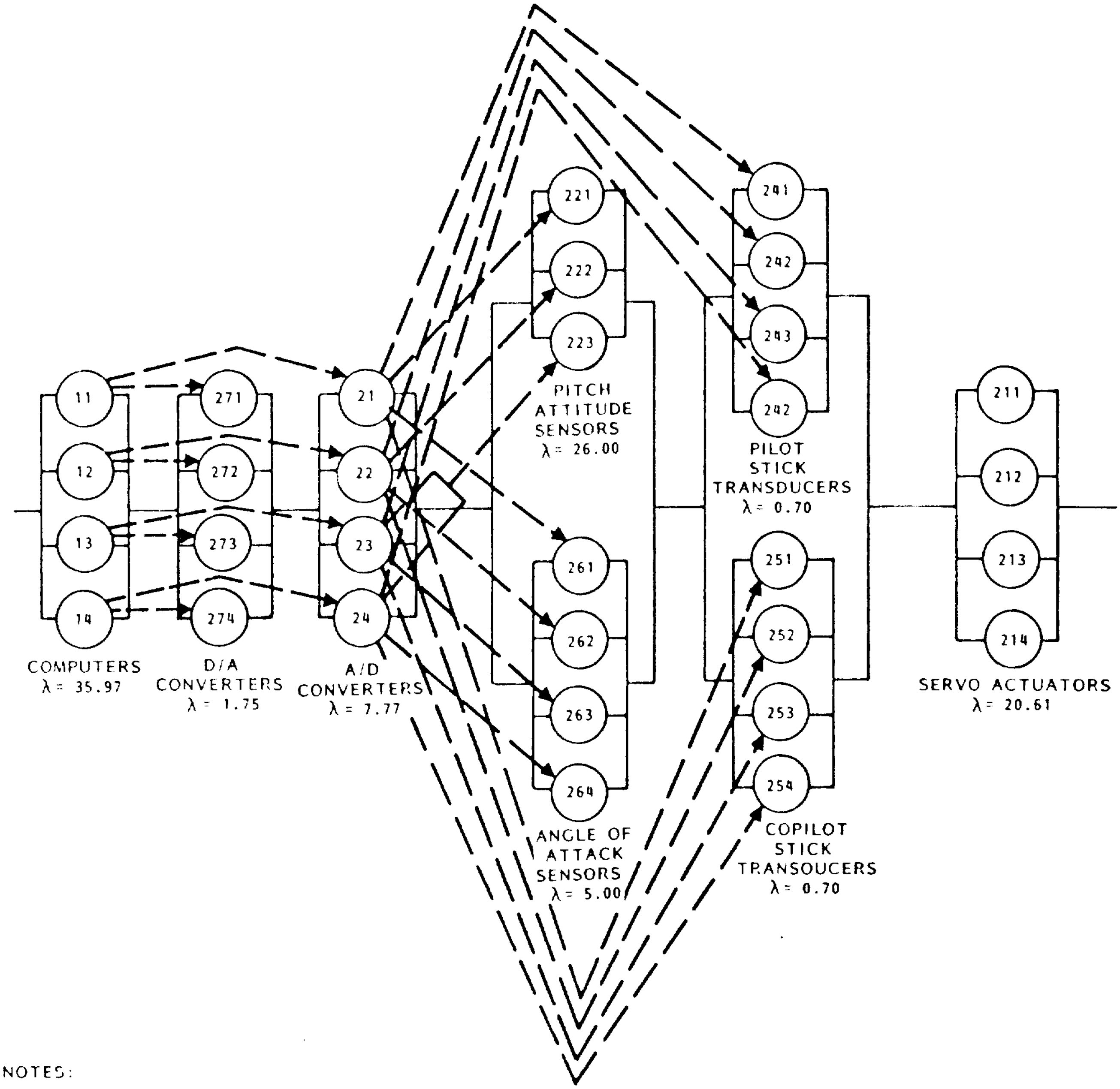
Tables 5 and 6 reveal flying qualities degradation as well as system failure, because the extent and likelihood of degradation are important measures of system acceptability. Cases 6 through 8 have been omitted because the straight FBW function does not degrade in stages; it in general fails completely when an appropriate combination of failures have occurred. Since stability augmentation degradation is basically sensor related, this set of servo-oriented architecture variations are not fully useful in delineating flying qualities trade-offs.

Certain factors, however, are rather interesting. Cases 11 and 12 show only a modest increase in flying qualities degradation for the fully triplex architecture, but notable disposition toward system failure. This suggests that the triplex computers are a weaker point in the configuration 12 than the three AOA pairs. Case 8 in Table 4 tends to reinforce this inference.

Table 4. DFCS Reliability Assessments

CASE	SYSTEM	SAS BACK-UP	SERVO SET-UP	SYSTEM FAILURE	
				5-HR	10-HR
1	Quad AFBW	Pitch Hold	Quad	.478 x 10 exp -10	.382 x 10 exp -9
2	Quad AFBW	None	Quad	.615 x 10 exp -10	.492 x 10 exp -9
3	Quad AFBW	None	Triplex	.626 x 10 exp -10	.500 x 10 exp -9
4	Quad AFBW	None	Dual-Dual	.107 x 10 exp -7	.430 x 10 exp -7
5	Dual-Dual AFBW	None	Dual-Dual	.200 x 10 exp -6	.798 x 10 exp -6
6	Quad FBW	N/A	Quad	.457 x 10 exp -10	.365 x 10 exp -9
7	Quad FBW	N/A	Triplex	.468 x 10 exp -10	.374 x 10 exp -9
8	Triplex FBW	N/A	Triplex	.153 x 10 exp -6	.612 x 10 exp -6
9	Quad SAS	Pitch Hold	Quad	.478 x 10 exp -10	.382 x 10 exp -9
10	Quad SAS	None	Quad	.615 x 10 ex -10	.491 x 10 exp -9
11	Quad SAS	None	Triplex	.626 x 10 exp -10	.500 x 10 exp -9
12	Triplex SAS	None	Triplex	.188 x 10 exp -6	.751 x 10 exp -6

ORIGINAL PAGE IS
OF POOR QUALITY



NOTES:

1. DASHED LINES WITH ARROWS INDICATE UNIDIRECTIONAL DEPENDENCIES
2. FAILURE RATES GIVEN IN FAILURES PER MILLION FLYING HOURS
3. BUS AND CONNECTOR FAILURES INCLUDED

Figure 12. DFCS Reliability Block Diagram

Table 5. Flying Qualities Degradation for a 5-Hour Mission

CASE	PROBABILITY OF FLYING QUALITIES LEVEL			
	1	2	3	Less than 3
1	.999+	.212 x 10 exp -6	.017 x 10 exp -11	.478 x 10 exp -10
2	.999+	.212 x 10 exp -6	0	.615 x 10 exp -10
3	.999+	.212 x 10 exp -6	0	.626 x 10 exp -10
4	.999+	.212 x 10 exp -6	0	.607 x 10 exp -10
5	.999+	.120 x 10 exp -3	0	.200 x 10 exp -6
9	.999+	.212 x 10 exp -6	.137 x 10 exp -10	.478 x 10 exp -10
10	.999+	.212 x 10 exp -3	0	.615 x 10 exp -10
11	.999+	.212 x 10 exp -6	0	.626 x 10 exp -10
12	.999+	.221 x 10 exp -6	0	.188 x 10 exp -6

ORIGINAL PAGE IS
OF POOR QUALITY

Table 6. Flying Qualities Degradation for a 10-Hour Mission

CASE	PROBABILITY OF FLYING QUALITIES LEVEL			
	1	2	3	Less than 3
1	.999+	.848 x 10 exp -6	.250 x 10 exp -14	.382 x 10 exp -9
2	.999+	.848 x 10 exp -6	0	.492 x 10 exp -9
3	.999+	.848 x 10 exp -6	0	.500 x 10 exp -9
4	.999+	.848 x 10 exp -6	0	.430 x 10 exp -7
5	.999+	.240 x 10 exp -3	0	.798 x 10 exp -6
9	.999+	.819 x 10 exp -6	.108 x 10 exp -9	.382 x 10 exp -9
10	.999+	.848 x 10 exp -6	0	.492 x 10 exp -9
11	.999+	.848 x 10 exp -6	0	.500 x 10 exp -9
12	.999+	.883 x 10 exp -6	0	.751 x 10 exp -6

4.2.1 Additional Quadruplex Architectural Variations

The foregoing architecture constitutes an older vintage of DFCS such as those retrofitted on an airplane originally wired for analog systems; this was a constraint imposed by the RDFCS architecture, which was similar to the L-1011 DFCS. In retrofit type situations, system interconnect wiring is usually dedicated point-to-point signal paths, with little use of digital MUX buses. Currently, parallel MUX buses are in common use, and considerably more complex buses topologies are expected for future system. Specifically, bandwidth, damage tolerance, and system integration are likely to motivate more divesity among MUX bus organizations.

4.3 Relaxed Static Stability Airplane

A negative static stability margin was postulated as a requirement for this investigation, and the existing flight cases for the NASA Ames RDFCS simulation (Reference 17) were altered to yield six RSS flight cases. The resultant RSS flight cases were then analyzed to determine the pitch axis dynamic behavior, and a non-realtime simulation was developed and checked against the predicted pitch-axis dynamics for the free or unaugmented airplane. The same flight case data were then used in the RDFCS facility simulation. Next, the non-realtime simulation time history responses were used to check the RDFCS simulation.

4.3.1 Airplane Simulations

Both the non-realtime and the RDFCS simulations were implemented using a state variable approach as depicted in Figure 13. Here the pitch-axis dynamics of the free RSS airplane, as captured in the matrix A, are quite divergent or unstable, so certain states must be fed back to enable a stability augmentation function. The state variables here are pitch, pitch rate, vertical axis velocity, and horizontal axis velocity. The first two states are inertially oriented, and are directly measured by airplane sensors. The second two are referenced to the airstream incident on the airplane, and are combined to form the directly measured signals, angle-of-attack and true airspeed. These two air data signals are produced using matrix C.

The above sensor feedback signals then appear in vector u, and the pilots' stick inputs are applied through vector d in Figure 13. No outer loop sensors as for autoland were used in this investigation, although they were included in the RDFCS simulation.

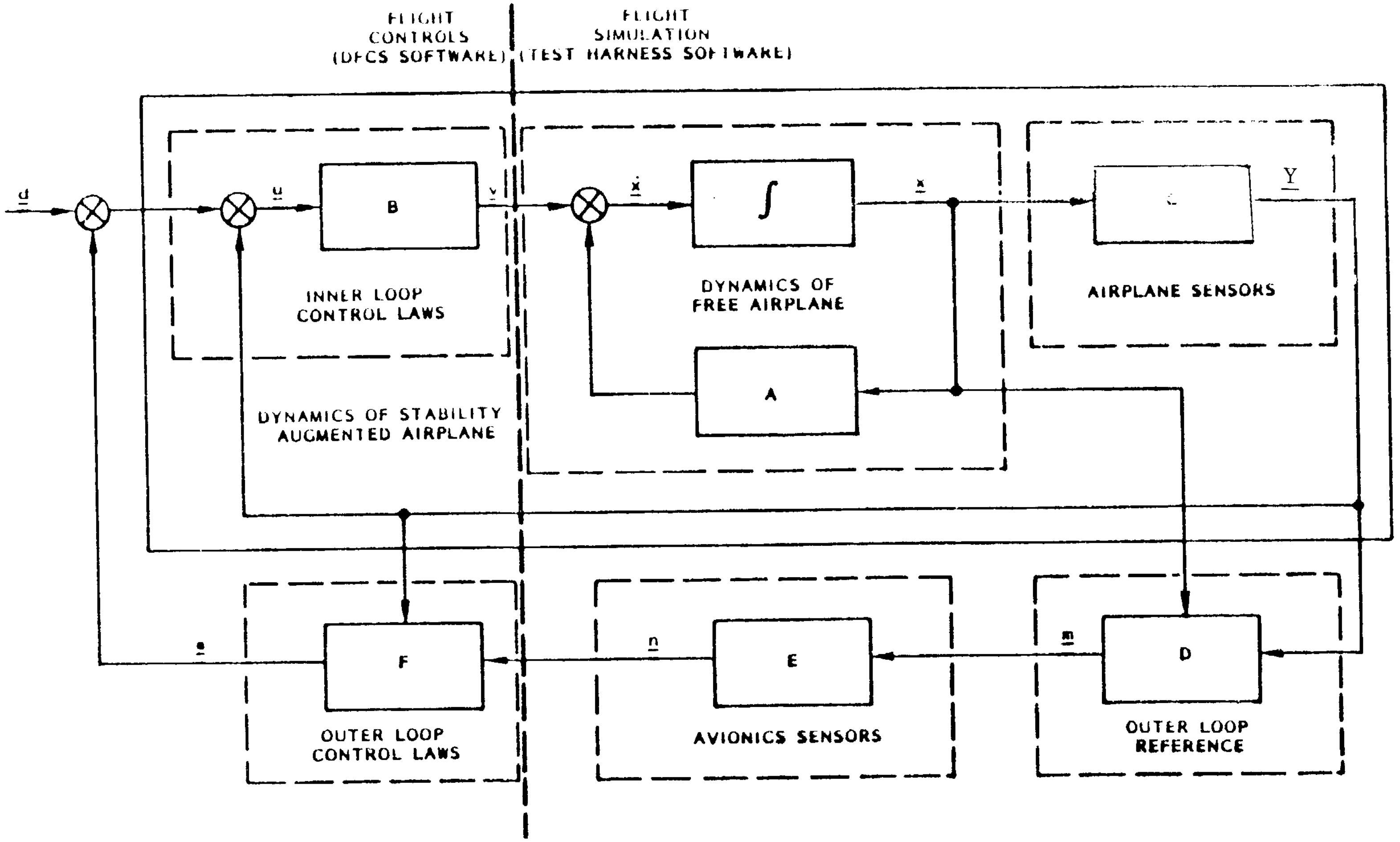

FLIGHT CONTROLS (DFCS SOFTWARE)
FLIGHT SIMULATION (TEST HARNESS SOFTWARE)
d
u
B
v
ẋ
∫
x
C
Y
INNER LOOP CONTROL LAWS
DYNAMICS OF FREE AIRPLANE
AIRPLANE SENSORS
A
DYNAMICS OF STABILITY AUGMENTED AIRPLANE
e
F
n
E
m
D
OUTER LOOP CONTROL LAWS
AVIONICS SENSORS
OUTER LOOP REFERENCE

Figure 13. Airplane Simulation Block Diagram

4.3.1.1 Flight Cases

Table 7 summarizes conversions of six existing basic wide-body transport type flight cases to six corresponding RSS flight cases with -5% static stability margins. For Flight Case A1RSS for example, calculations revealed the neutral point to be at 53% of the mean aerodynamic chord (MAC), where neutrality denotes no pitch moment change with a change in angle-of-attack. More specifically, the stability derivative describing the airframe behavior goes to zero. The -5% static margin then corresponds to shifting the center-of-gravity aft to 58% MAC.

This type of alteration to the airframe dynamics yields a pair of real or non-oscillatory roots, "Tau 1" and "Tau 2" in Table 7. Note that the negative time constants for Tau 2 correspond to positive roots, which plot on the positive axis of the s-plane, produce an absolute instability or exponential divergence that dominates the dynamic response of the airplane's pitch-axis. With a negative 5% margin, moreover, the rate of divergence is rather rapid, and this is clearly unacceptable For Flight Case F6RSS, for example, the 3.17 second time constant yields a doubling of pitch attitude in 2.2 seconds, which is quite rapid and unflyable.

Such a tendency must be overcome by a stability augmentation function whose closed-loop eigenvalues rectify these kinds of dynamic responses. Note that instability per se is not necessarily unacceptable for four of the basic flight cases exhibit a negatively damped phugoid. In all cases, however, the negative damping is quite small and the phugoid period is relatively long.

4.3.1.2 Simulation Organization

A non-realtime simulation was used to generate airplane time history check cases prior to work at the RDFCS simulator. The organization of the airplane simulation was actually the same as that of the one which had previously been installed in the PDP-11/60 at NASA Ames. For the RSS flight cases, no changes were made to the software organization itself, however, other than to add interfaces for the SAS control laws. Basically, the organization of the simulation provides for the conversion of flight case data into a discrete-time model, trimming for specified initial conditions, and the generation of dynamic time history outputs. Ground effects, random gust options, or point simulation flight case transitions may be selected.

Table 7. Relaxed Static Stability Flight Cases

FLIGHT CASE	CENTER OF GRAVITY % $\bar{c}$	NEUTRAL POINT % $\bar{c}$	V_T fps	τ_1 sec	τ_2 sec	ω_{PH} rad/sec	ζ_{PH} -	$\frac{\Delta C_M}{\Delta \delta_H}$ 1/rad	$\frac{\Delta C_M}{\Delta q}$ 1/rad	$\frac{\Delta C_M}{\Delta \dot{\alpha}}$ sec/rad	$\frac{\Delta C_M}{\Delta C_L}$ -	$\frac{\Delta C_M}{\Delta \alpha}$ 1/rad
A1RSS	58.0	53.0	283.7	1.099	-5.86	0.169	0.276	-3.247	-11.48	-3.14	0.050	0.295
C13RSS	51.0	46.0	910.7	0.160	-2.40	0.008	0.325	-2.830	-14.3[illegible]	-7.83	0.044	0.353
C15RSS	54.2	49.2	442.2	0.122	-1.64	0.134	0.526	-2.415	-12.15	-3.44	0.053	0.288
D2RSS	52.8	47.8	487.1	0.660	-7.55	0.125	0.453	-2.533	-12.75	-3.18	0.050	0.292
E3RSS	55.5	47.8	265.7	0.938	-7.09	0.167	0.488	-3.138	-11.76	-3.14	0.053	0.306
F6RSS	54.4	49.4	224.1	0.920	-3.17	0.209	0.192	-3.016	-11.23	-3.03	0.170	0.960

ORIGINAL PAGE IS
OF POOR QUALITY

4.3.2 Flight Case Analyses

Two stages of analysis were performed. First, the RSS airplane behavior was approximated by aft shifting of the center-of-gravity to -5% MAC. Note that six old flight cases have been converted into six derivatives sensitive to the reduced lever arm of the empennage about the new center of gravity ere appropriately changed. As indicated in Table 7, these all relate to the generation of pitching moment. The analysis of the RSS flight cases then involved the examination of their respective dynamics. Specifically, the RSS stability derivatives ere used to calculate the free airplane response by finding the root solutions to the characteristic equation for each flight case. These results also are given in Table 7.

Basically, defining a RSS flight case involves finding the neutral point center-of-gravity, i.e., the point at which no pitching moment results from a lift change on the wing. This point, in terms of percent of the mean aerodynamic chord (MAC), is therefore found to be 53% for Flight Case AlRSS. Since a -5% stability margin is desired, the center-of-gravity for Flight AlRSS 58%. The shortened lever arm from the horizontal stabilizer to the center-of-gravity produces a proportionate reduction in moment generating capability, as evident in certain of the RSS stability derivatives given in Table 7.

Note that the sign on pitching moment due to wing lift changes from "minus" for 25% to "plus" for 58%. In the RSS case then, increasing lift produces more nose up moment, which in turn increases lift further. This positive feedback effect constitutes unconditional instability in the free or unaugmented airplane response. The rate of divergence is quite rapid. For Flight Case F6RSS, for example, the time constant of -3.17 seconds yields a doubling of pitch attitude every 2.2 seconds. The given time constants, radial frequencies, and damping ratios were obtained from the root solutions of the characteristic equations for the respective flight cases, as determined by the stability derivatives.

4.3.2.1 Simulation Check Cases

Simulation check cases were first run using the non-realtime dynamic simulation. Time histories were generated for each case, and checked against the root solutions. For the RSS cases, the divergent real root dominated the time histories. Hence, the pitch angle sequence exhibits an exponential time constant roughly the same as the analytical calculated time constant value of -3.17 seconds for Flight Case F6RSS once the initial transient has subsided. The initial upset employed here of 3.0 degrees/second of pitch rate is not really needed to exhibit the instability. Rather, it is used as a standard upset as needed for the corresponding augmented airplane.

While these free airplane responses are used as a cross-check on the real-time airplane simulation, the augmented airplane response is used to check both the simulation and the stability augmentation control laws. Analytical root solution (eigenvalue) checks can also be made for the augmented airplane response, wherein the order of the characteristic equations is increased due to the presence of the sensors feedbacks. All these checks should be in general agreement, and they serve to corroborate the control law done in different stages of development.

4.3.2.2 Stability Augmentation Requirements

In general, the requirements for stability augmentation are to effect desirable flying qualities for the augmented airplane. Basically, the eigenvalues of the closed-loop airplane should exhibit suitable damping and frequency characteristics. In effect, the augmentation control law should override the RSS airplane's positive pitching moment due to increasing wing lift. Restoration of the negative pitching moment is possible through a nose-down stabilizer input for increasing angle-of-attack.

For the subject demonstration, the design criteria was simply to provide approximately the same, or better, pitch axis handling and response for the stability augmented RSS augmented airplane as was available on the basic 25% MAC center-of-gravity free airplane.

4.4 RDFCS Facility Modifications

Because of the scope and complexity of the RDFCS simulator, the changes to the facility were rather extensive. Converting a single fail-operational system to a double fail-operational one is clearly non-trivial, and altogether the details handled were quite appreciable. The DFCS software, the airplane simulation, and the system interconnection were re-worked. Then the test software to evaluate the AFBW system were developed. With respect to Figure 14, software changes or additions were made to the FCCs, the MDICU, and the PDP-11/04. Some new pin assignments were made for the back connector breakout panel pins, and several new switch functions were designated at the logic discrete panel.

4.4.1 Pitch-Axis Augmented Fly-by-Wire System

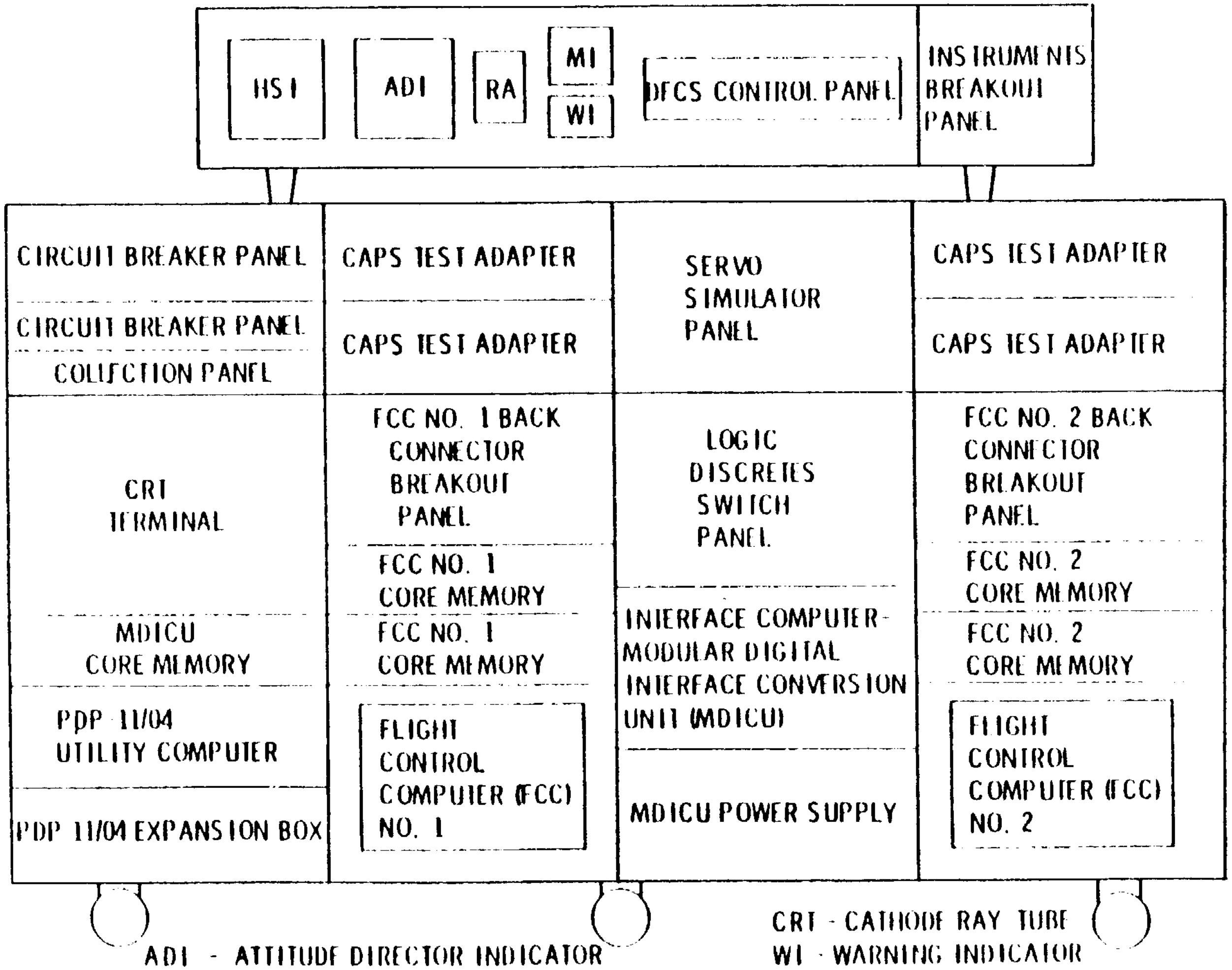
Basically, an essentially new DFCS flight software load module was developed. A double fail-operational system architecture was implemented with computational frame synchronization across the four channels. Channel coordination was accomplished using control variables broadcasted by each channel over existing serial digital buses. This involved new absolute address assignments during software linking. As a result, the AFBW was implemented without any hardware modifications to the flight computers. Because of the relatively slow refresh rate of the asynchronous digital buses, the frame synchronization process was much slower than customary. But the effect on system operation was not consequential.

4.4.1.1 Stability Augmentation Control Laws

The control laws were implemented using the customary Tustin transform, with variable scaling to unity. The details for any one channel appear in Figure 15. Only the pilot's stick was connected, so there was no stick blending logic. The voter/comparator thresholds are given in Table 8.

4.4.1.2 Augmented Airplane Check Cases

The dynamic response of Flight Case F6RSS for the free airplane simulated in the PDP-11/60 is presented in Figure 16. An initial pitch rate upset is introduced, and the ensuing pitch axis divergence was found to conform to the non-realtime check case time history. Noting the trace for pitch attitude excursions about trim, a time constant of about 3.0 seconds can be observed over seven one-second intervals averaged between 1 and 8 seconds. This compares with the computed positive real root in Table 6 of 3.17 seconds. All six RSS flight cases were checked out in this same manner, comparing time histories and calculating response time constants.



ADI	- ATTITUDE DIRECTOR INDICATOR	CRT	- CATHODE RAY TUBE
CAPS	- COLLINS ADAPTIVE PROCESSOR SYSTEM	WI	- WARNING INDICATOR
HSI	- HORIZONTAL SITUATION INDICATOR	MI	MODE ANNUNCIATOR INDICATOR
		RA	- RADAR ALTIMETER INDICATOR

Figure 14. Palletized DFCS

ORIGINAL PAGE IS OF POOR QUALITY

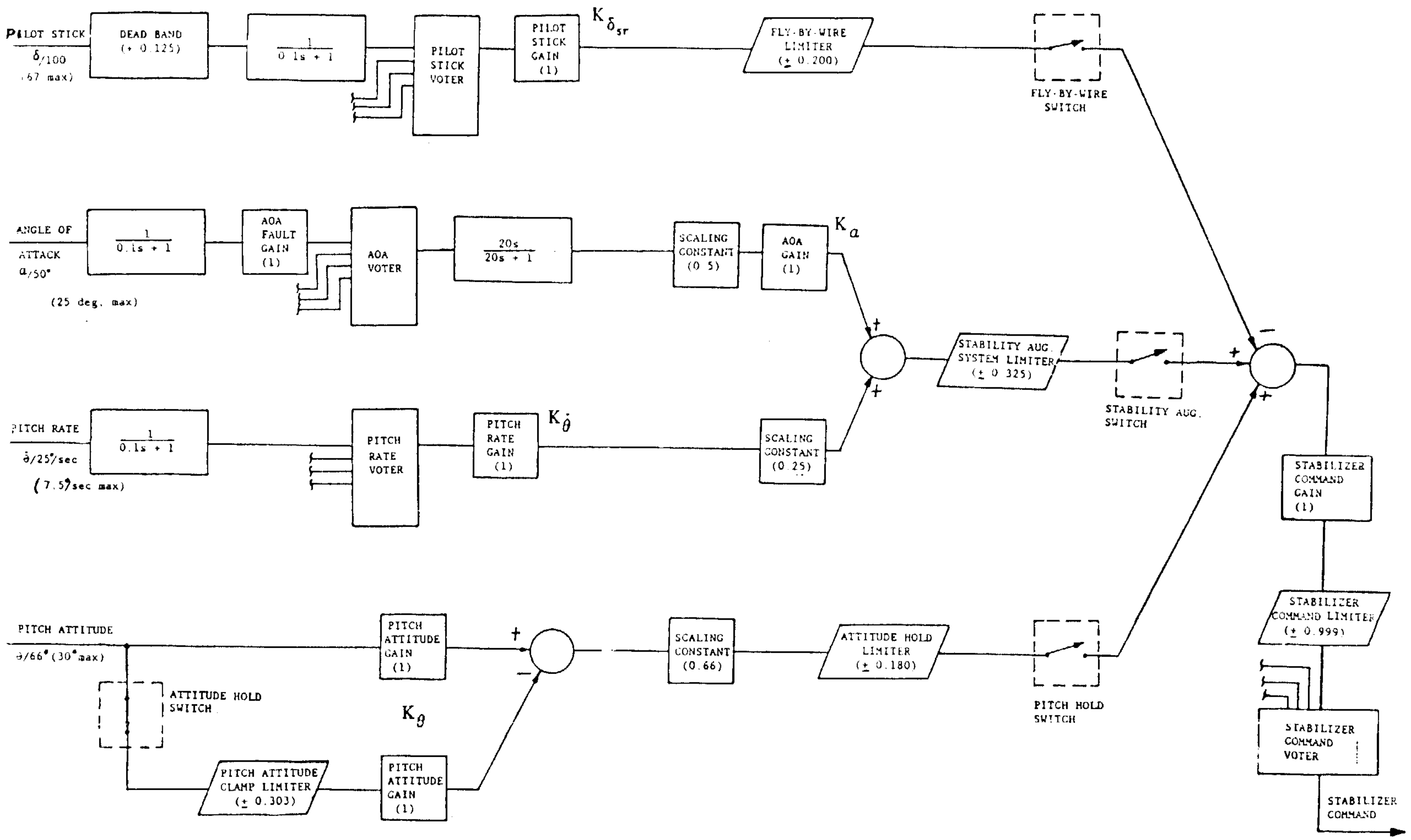


Figure 15. Control Law Block Diagram

ORIGINAL PAGE IS OF POOR QUALITY

Table 8. Sensor Comparator Thresholds

MONITORED SIGNAL	COMPARATOR		VALIDITY FLAG	
	AMPLITUDE THRESHOLD	TIME DELAY	TRIP DELAY	HEALING DELAY
Angle-of-Attack	1.0 degrees	250 ms	1.0 sec	2.0 sec
Pilots' Stick Inputs	5% full stick	200 ms	N/A	N/A
Pitch Rate	0.5 deg/sec	400 ms	250 ms	1.0 sec
Stabilizer Command	0.1 degrees	100 ms	100 ms	100 ms

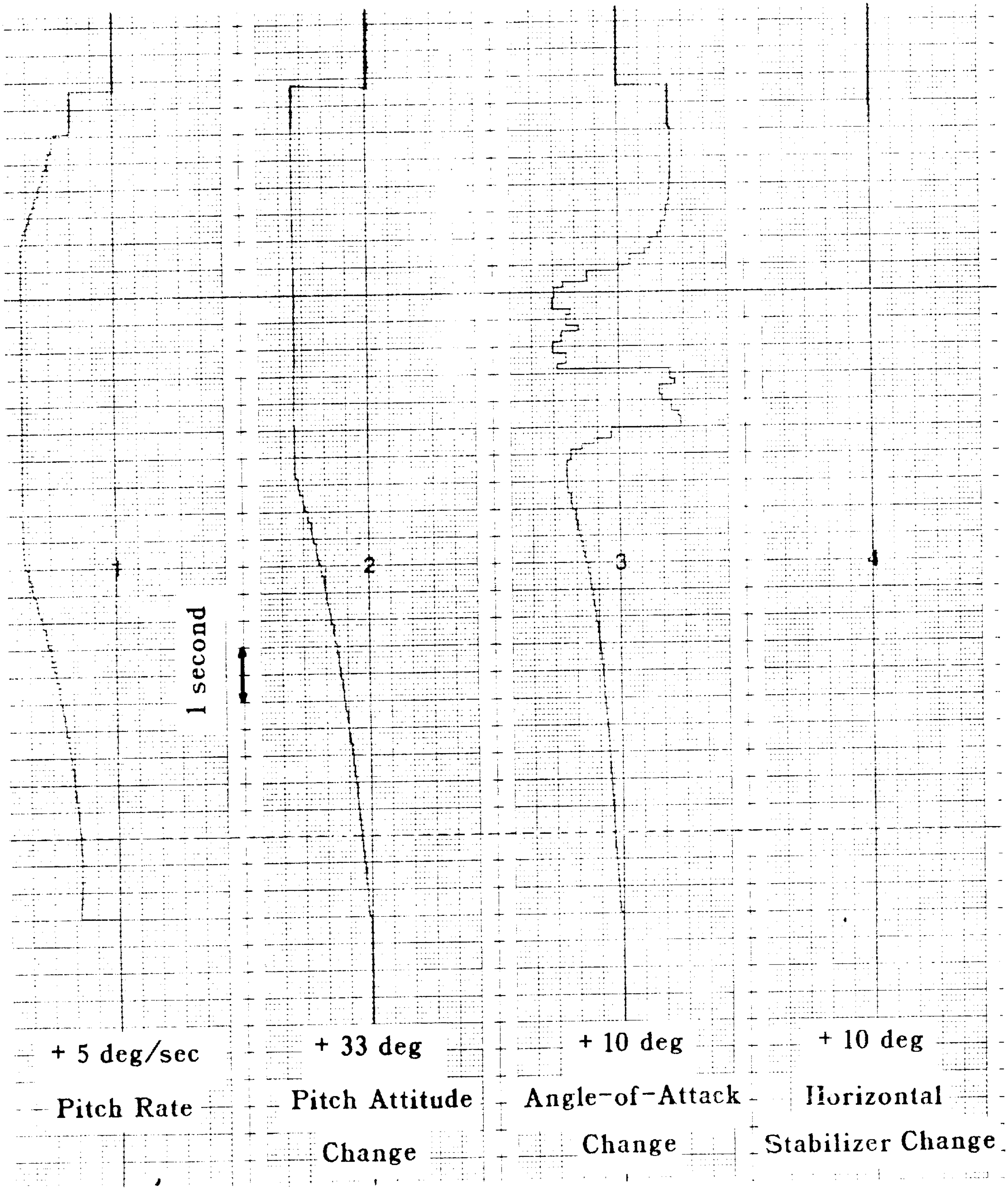


Figure 16. Free RSS Airplane Time History

ORIGINAL PAGE IS
OF POOR QUALITY

4.4.2 Augmented Fly-by-Wire Modifications

Simulated interconnect wiring changes are noted in Tables 9 and 10. Sheet 1 of Table 9 summarizes how the simulation output was modified to provide the sensor signals needed for the SAS function. Sheet 2 shows how these signals were directed through the MDICU and into the FCCs. In all, three software programs had to be altered to effectuate these changes. Table 9 summarizes the discrete logic input signals and their routing into the FCCs. Note that FCC memory locations were chosen to be the same in each computer channel so that the software in each channel would be as well.

Figure 17 indicates the top-level software flow in the FCCs. This encompasses and implements the previously verified software design. Note that several levels of control logic appear in a flow chart form here, but there remains a one-to-one correspondence with the design. The flight control laws were calculated in the Foreground segment.

The closed-loop stability augmentation short-period response for Flight Case F6RSS is shown in Figure 18. The initial pitch rate is rather quickly damped out with only a small second overshoot. The augmentation command produces a smooth horizontal stabilizer input that vanishes in 5 seconds. The phugoid for the augmented airplane is not evident in the brief time history, but there are no indications of significant looseness or divergence. Obviously, the flying qualities of the augmented airplane here are quite good, and in the case of the other RSS flight cases as well. This caliber of pitch axis response then serves as the reference to which flying qualities degradation is assessed during sensor failure effects testing.

Table 9. Analog Signal Reassignments (Sheet 1 of 2)

PDP-11/60					
NEW SIGNAL	OLD SIGNAL	OLD/NEW PROGRAM	IBUFIO LOCATION	OLD SCALING	NEW SCALING
LH AOA	Lat. Accel.	LOADF/ SENSIO	229	128.8 fps	50 deg
" "	" "	"	222	"	"
RH AOA	Norm. Accel.	LOADF/ SENSIO	201	128.8 fps	50 deg
" "	" "	"	225	"	"
P. Rate	Yaw Rate	MDIERT/ MDIERT	242	66.3 o/sec	25 o/sec
" "	" "	"	241	"	"
" "	" "	"	234	"	"
Pitch	Pitch	MDIERT/ MDIERT	235	100 deg	100 deg
"	"	"	209	"	"
"	"	"	193	"	"

ORIGINAL PAGE IS
OF POOR QUALITY

Table 9. Analog Signal Reassignments (Sheet 2 of 2)

MDICU					FCC		
NEW SIGNAL	INPUT ADDRESS	TRUNK	GAIN	DEVICE	LRU NUMBER	PIN NUMBERS	MEMORY LOCATION
LH AOA	FF24	44	.99999	DAE(2)	1 & 2	P1A-23,30	FB10
" "	FF1D	33	"	DA(12)	"	P4A-23,30	"
RH AOA	FF08	8	"	DA(0)	1 & 2	P1A-22,35	FB1F
" "	FF20	40	"	DA(14)	"	P4A-22,35	"
P. Rate	FF29	49	"	WAC(1)	1	P1A-24,37	FB11
" "	FF31	61	"	WAC1(1)	2	"	"
" "	FF30	60	"	WAC1(0)	1 & 2	P4A-24,37	"
Pitch	FF00	0	.55556	DS(0)	1	P1A-18,30,31	FB03
"	FF10	20	"	DSE(0)	2	"	"
"	FF2A	50	"	DS(1)	1 & 2	P4A-18,30,31	"

ORIGINAL PAGE IS
OF POOR QUALITY

Table 10. Discrete Signal Reassignments

SWITCH LABEL		OLD PINS		NEW PINS		NEW	NEW MEMORY LOCATION		
NEW	OLD	FCC 1	FCC 2	FCC 1	FCC 2	CHANNEL	WORD	BIT	ADDRESS
LH AOA Val 1	AOA V1A	P3A-46	--	P2B-57	--	1	DI.W4	14	F84D
LH AOA Val 2	AOA V1B	P3A-37	--	P3B-57	--	2	"	"	"
RH AOA Val 1	AOA V2A	--	P3A-46	--	P2B-57	3	"	"	"
RH AOA Val 2	AOA V2B	--	P3A-37	--	P3B-37	4	"	"	"
P Rate Val 1	Yaw R V1A	P2A-64	--	P2B-58	--	1	DI.W4	15	F84D
P Rate Val 2	Yaw R V1B	P3A-64	--	P3B-58	--	2	"	"	"
P Rate Val 3	Yaw R V2A	--	P2A-64	--	P2B-58	3	"	"	"
P Rate Val 3	Yaw R V2B	--	P3A-64	--	P3B-58	4	"	"	"
Pitch SAS On	SAS ON 1A	P2B-12	--	P2B-55	--	1	DI.W4	12	F84D
" " "	--	--	--	P3B-55	--	2	"	"	"
" " "	SAS ON 2A	--	P2B-12	--	P2B-55	3	"	"	"
" " "	--	--	--	--	P3B-55	4	"	"	"
Pitch Hold	--	--	--	P2B-56	--	1	DI.W4	13	F84D
" "	R NAV V1B	P3B-12	--	P3B-56	--	2	"	"	"
" "	--	--	--	--	P2B-56	3	"	'	"
" "	R NAV V2B	--	P3B-12	--	P3B-56	4	"	"	"
FBW On	--	--	--	P2B-46	--	1	DI.W4	11	F84D
" "	--	--	--	P3B-46	--	2	"	"	"
" "	--	--	--	--	P2B-46	3	"	"	"
" "	--	--	--	--	P3B-46	4	"	"	"

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS OF POOR QUALITY

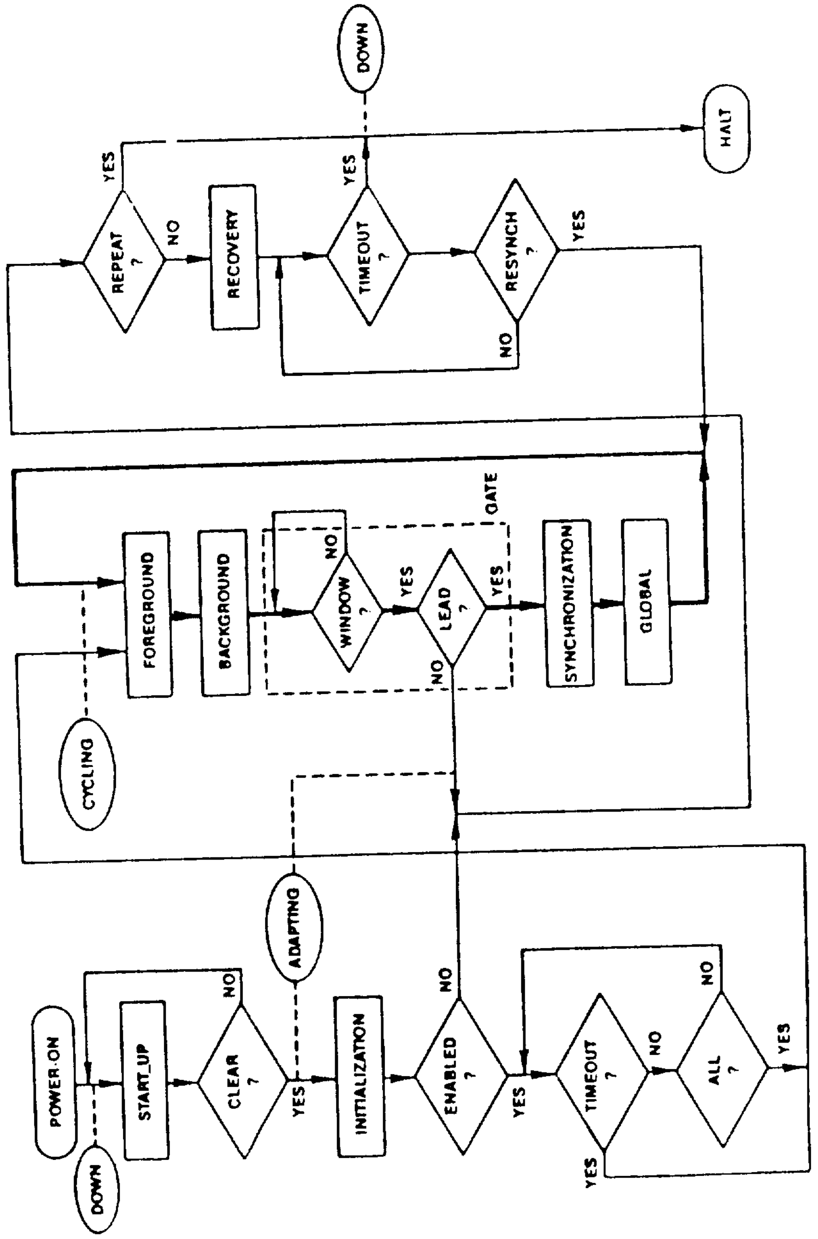


Figure 17. Software Control Flow Diagram

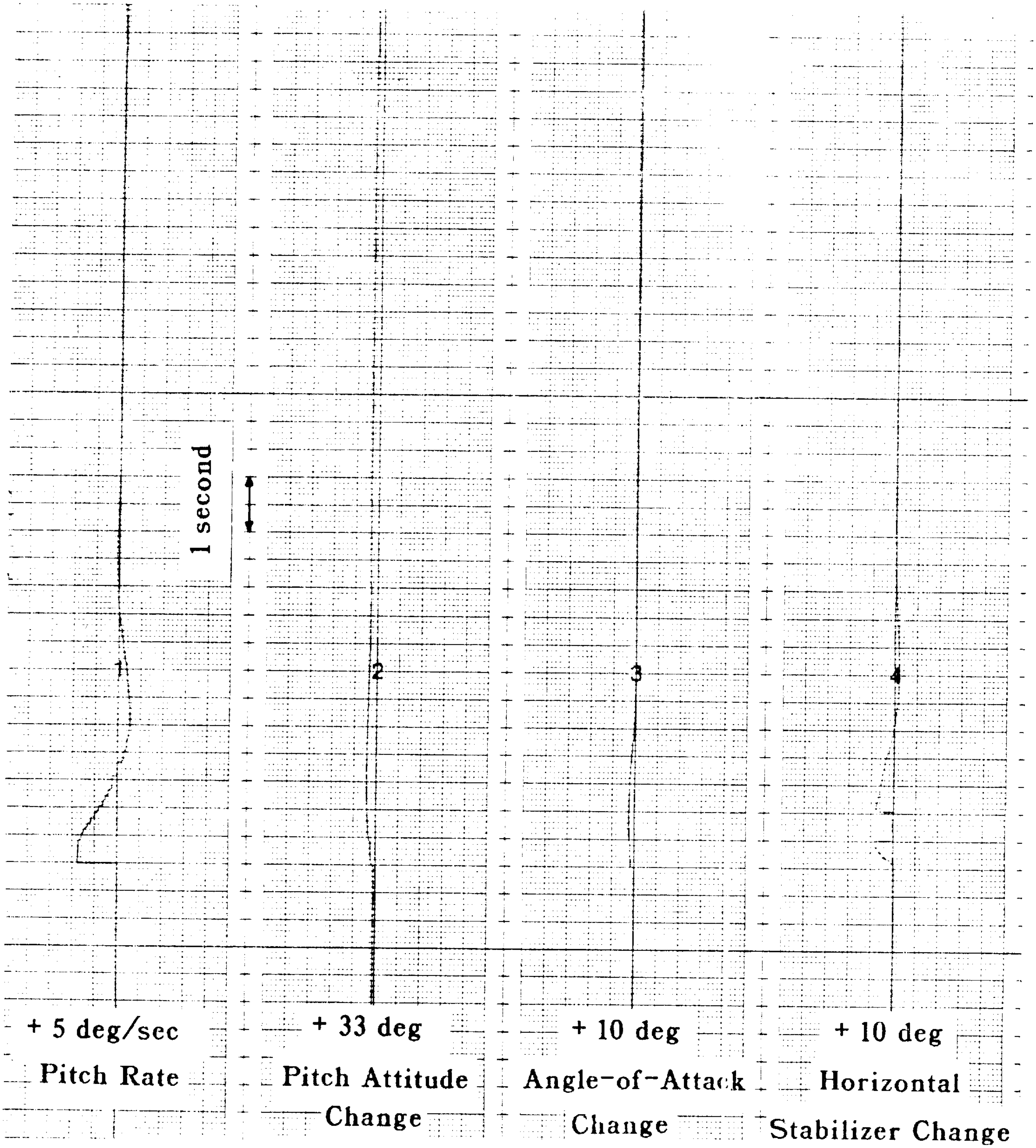


Figure 18. Augmented RSS Airplane Time History

ORIGINAL PAGE IS
OF POOR QUALITY

4.5 Simulator Investigations

The investigations performed on the AFBW system included: system-level failure effects testing that focused on flying qualities degradation; and real-time, multilevel assessment of software behavior that focused on architectural fault tolerance. The former type testing was rather conventional, and as noted earlier, impeded by a low-fidelity pilot interface. The latter was rather sophisticated, and is thought to constitute a valuable new testing methodology for system simulators.

4.5.1 Investigation Test Plan

The test plan focused on demonstrating multilevel testing as a means of obtaining higher confidence in the airworthiness of a DFCS. Table 10 indicates five levels of testing undertaken. The top level is conventional system or functional testing, as required here to examine RSS flying qualities. Four specific levels of the control structure were explored coincidentally. These were related to prior development activities as depicted in Figure 19. To accomplish this, there was appreciable emphasis on automated testing and wideband instrumentation. In all, these thrusts were seen as vital new ways of utilizing "ironbird" system simulators.

Table 11. DFCS Testing Scenario

TYPE OF TESTING	FOCUS	EXECUTION MONITOR	MONITOR LOCATION	PRIMARY CONCERN
SYSTEM VALIDATION	SYSTEM PERFORMANCE	NONE	N/A	FAULTED FLYING QUALITIES
SYSTEM VERIFICATION	SYSTEM STATE	NESTED FINITE-STATE MACHINES	SIMULATION/ TEST COMPUTER	FAULTED STATES
	CHANNEL SYNCHRONIZATION	PREDICATE/ TRANSITION NETWORK	SINGLE CHANNEL	SINGLE-POINT FAILURES
	MODE/FAULT LOGIC	BOOLEAN EXPRESSIONS	SINGLE CHANNEL	LOGIC CORRECTNESS
	CONTROL FLOW	CONTROL GRAPHS WITH ASSERTIONS	SINGLE CHANNEL	PATH TRAVERSALS

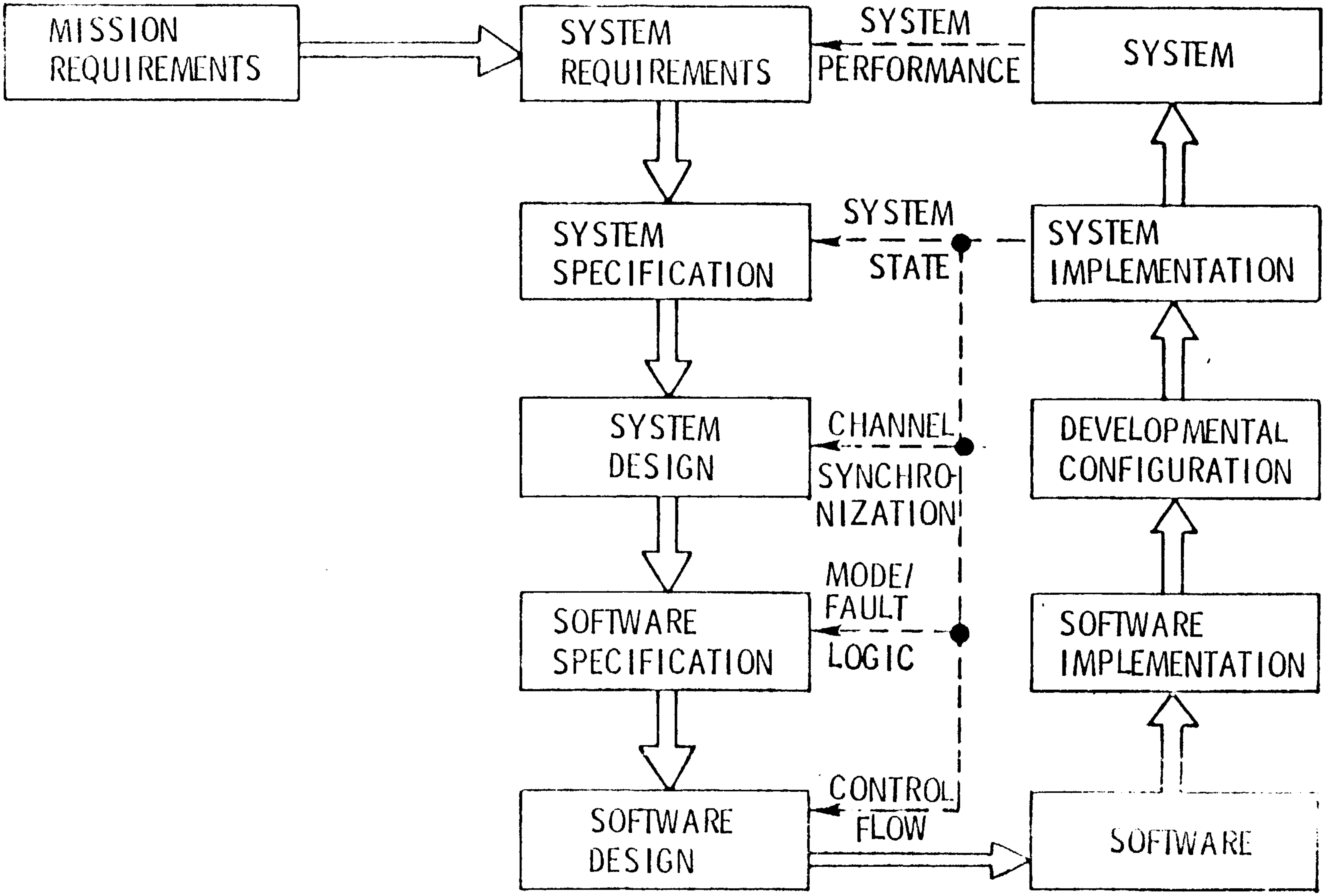


Figure 19. Multilevel Testing Closure

4.5.2 Test Execution Monitor

Two test execution monitors were developed and demonstrated during the AFBW investigations. The scenario employed is represented in Figure 20. The successful use of these real-time execution monitors was a major accomplishment of this investigation. System state logic was observed by an execution monitor in the PDP-11/60, via an instrumentation link through the PDP-11/04. The execution monitor was basically a finite-state machine that was driven in parallel with the DFCS software by the same logic inputs. The monitor checked to see that the DFCS software and the PDP-11/60 software remained in accord with the design specified state of the system.

The second execution monitor resided in one of the flight computers because of PDP-11/04 instrumentation bandwidth limitations. This monitor focused on the three-way channel synchronization process, under the assumption of the prior loss of the fourth processor. Specifically, the synchronization control signals for the other three channels were observed and compared with the nested state transition graphs of Figure 7. Note that this same form of test criteria was applicable to the predicate/transition network simulation described previously. This precise means of propagating accountability and ensuring consistency is considered a superior and quite practical way of fostering a quality DFCS.

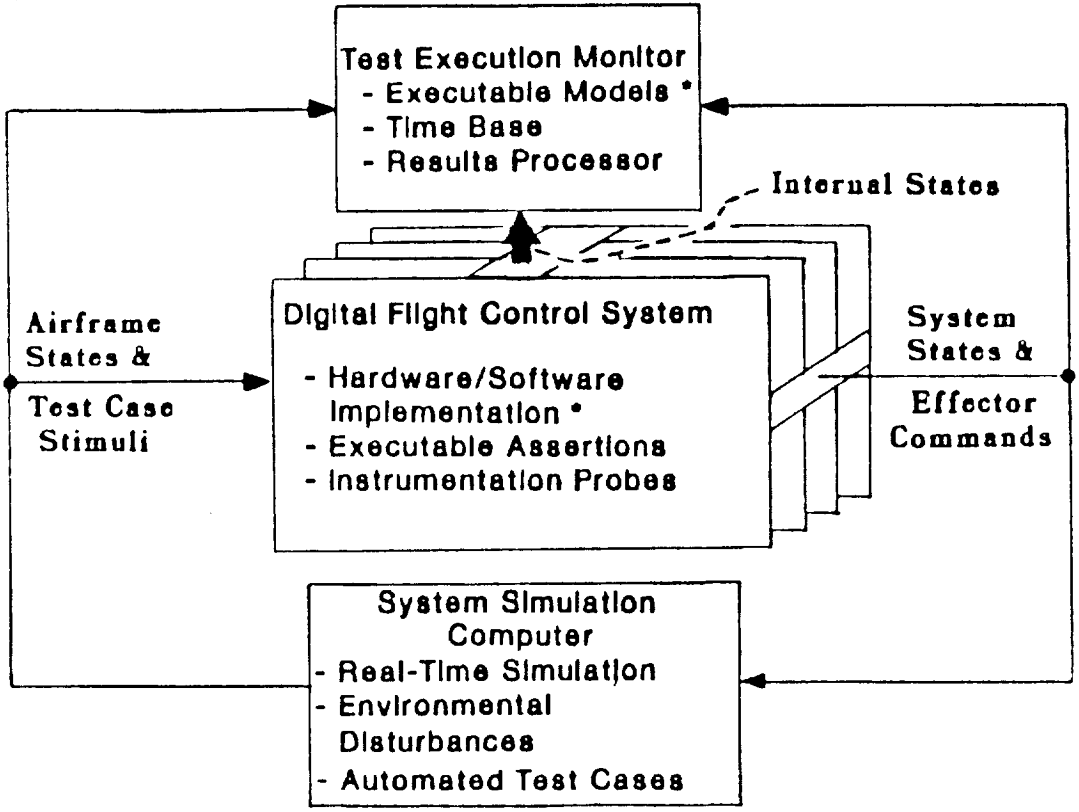


Figure 20. Automated Testing Scheme

4.5.3 Simulator Testing

Actual testing of the quadruplex DFCS included checkout, development, and demonstration tasks. System failure effects testing focused largely on flight computer failure to validate architectural fault tolerance, and sensor failure effects to validate graceful degradation of flying qualities. The conduct of multichannel synchronization was captured primarily in the time history form shown in Figure 21, and on a more detailed level, by the real-time execution monitors.

Each pulse in Figure 21 corresponds to a 50 millisecond computational frame, and the pulse amplitude denotes which of four foreground executive program control paths is being traversed. Here cross-channel frame and path synchronization is being maintained. Faults or transient disruptions were simulated by halting a processor at its CAPS test adapter panel, or by interrupting electrical power at the circuit breaker. Flying qualities degradation was observed by upon simulating multiple sensor faults for a given type of sensor, pitch rate or angle-of-attack. Sensors faults were applied through the MDICU or at the back connector breakout panel.

4.5.3.1 Synchronization Failure Effects

Figure 22 shows the slowed-down initial synchronization of three DFCS channels. At the outset, only Channel 3 is cycling in the foreground mode. Channel 2 then synchronizes, followed by Channel 1. Here Channel 4 is being used to run a real-time execution monitor, so it cannot come on line. Other exercises involved drop-out and re-synchronization as individual computer channels were halted and released. Re-synchronization always occurred within two cycles.

By chance, an actual clock tolerance problem occurred in one DFCS channel during the early stages of testing, thereby causing frequent synchronization drop-out of the affected channel. At first, a DFCS software flaw was suspected, but it was eventually determined that the hardware was at fault. The re-synchronization software, moreover, was responding promptly and correctly.

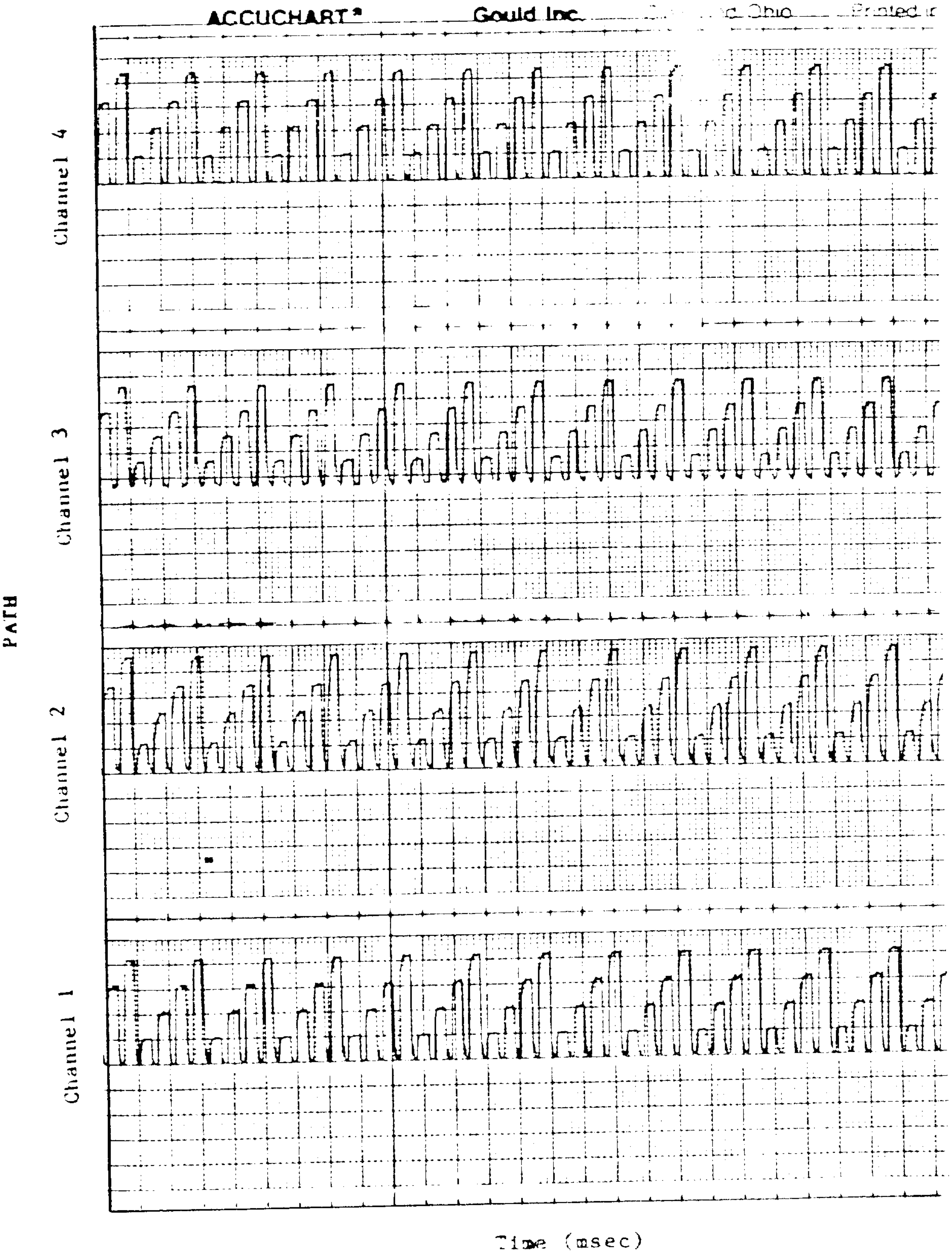


Figure 21. Normal Channel Synchronization Time History

ORIGINAL PAGE IS
OF POOR QUALITY

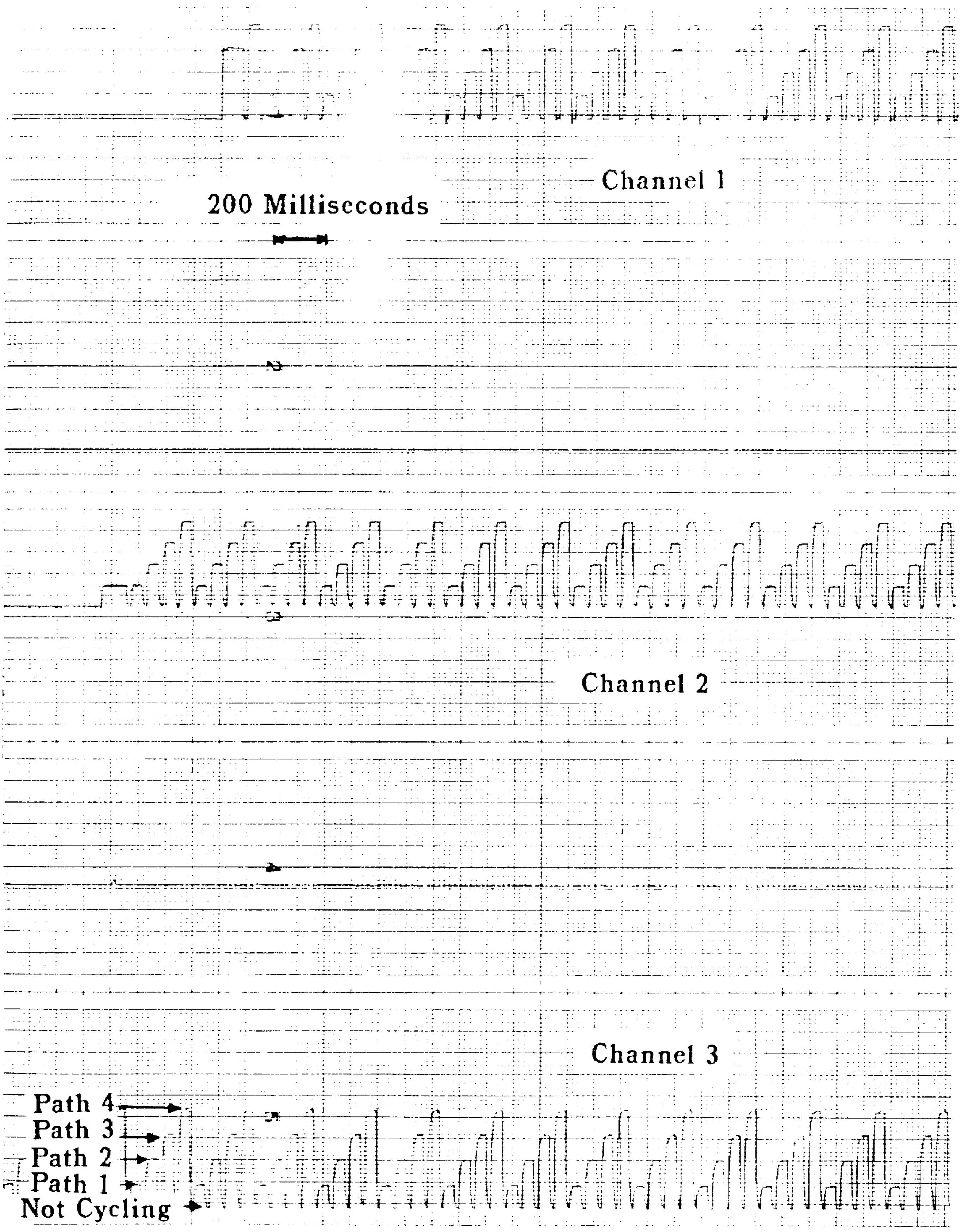


Figure 22. Start-Up Channel Synchronization Time History

ORIGINAL PAGE IS
OF POOR QUALITY

During subsequent multiple computer channels shutdown/re-start testing, a re-synchronization discrepancy was noted. Specifically, the logic for getting two synchronized pairs on different foreground executive paths was found to be inadequate. Basically, each pair remained synchronized, but the two pairs would not synchronize with each other. Representing the same conditions in the predicate/transition network simulation revealed the same problem, so the design had been incompletely defined, and the design verification test cases had failed to discern this. Probably, a design verification test case walkthrough would have strengthened the simulation testing so as to reveal the problem early-on. Nonetheless, it was gratifying and instructive to note that the design verification simulation had the inherent modeling power to characterize and resolve the problem seen in system simulation.

4.5.3.2 Sensor Failure Effects

In Figure 23, pitch rate sensors faults have been simulated to to remove their contribution from the stability augmentation stabilizer command. With the same initial pitch rate as used in Figure 16, the pitch-axis short-period response is seen to be somewhat degraded. The amplitude stabilizer input is noticeably less in Figure 23 than in Figure 17, but the persistence in the former case is more extended. The result a less damped, more sluggish response that constitutes the flying qualities degradation for the multiple pitch rate fault case. Still, the flying qualities are not unsafe. Further analysis, and probably pilot-in-the-loop simulation would be needed to determine the extent of the degradation. Also, the full set of relevant RSS flight cases would have to be so evaluated to identify the worst case pitch rate sensor fault situation.

Loss of all angle-of-attack sensor feedback is represented in the time history response in Figure 24. The flying qualities degradation here is very much worse than that shown in Figure 23, as had been recognized in the early stages of development and in the reliability analysis. Still, Figure 24 reveals some degree of benefit resulting from the pitch rate feedback as contrasted with the free airplane response in Figure 16. The horizontal stabilizer corrections in Figure 24 are attributable to pitch rate feedback, but the pitch attitude divergence nonetheless occurs, albeit at a more controlled rate than in Figure 16. Although pilot-in-the-loop assessment was not possible, the degree of flying qualities degradation here is deemed to be unacceptable and unsafe regarding the difficulty of manually flying such an airplane.

Note that the autopilot attitude hold mode provides a back-up capability for managing the pitch axis divergence, but the problem then is the inconvenience in maneuvering the airplane as desired. A pitch command knob or a pitch-axis control wheel steering input might be used, but this may not be very acceptable for extended flight. In such a situation, the assessment of safety and flying qualities must be carefully performed on a special case bases, where the system fault management strategy and operational procedures have been incorporated into the overall DFCS design. The same basic assessment approach, however, is still employed, namely that of a phased, integrated assurance methodology.

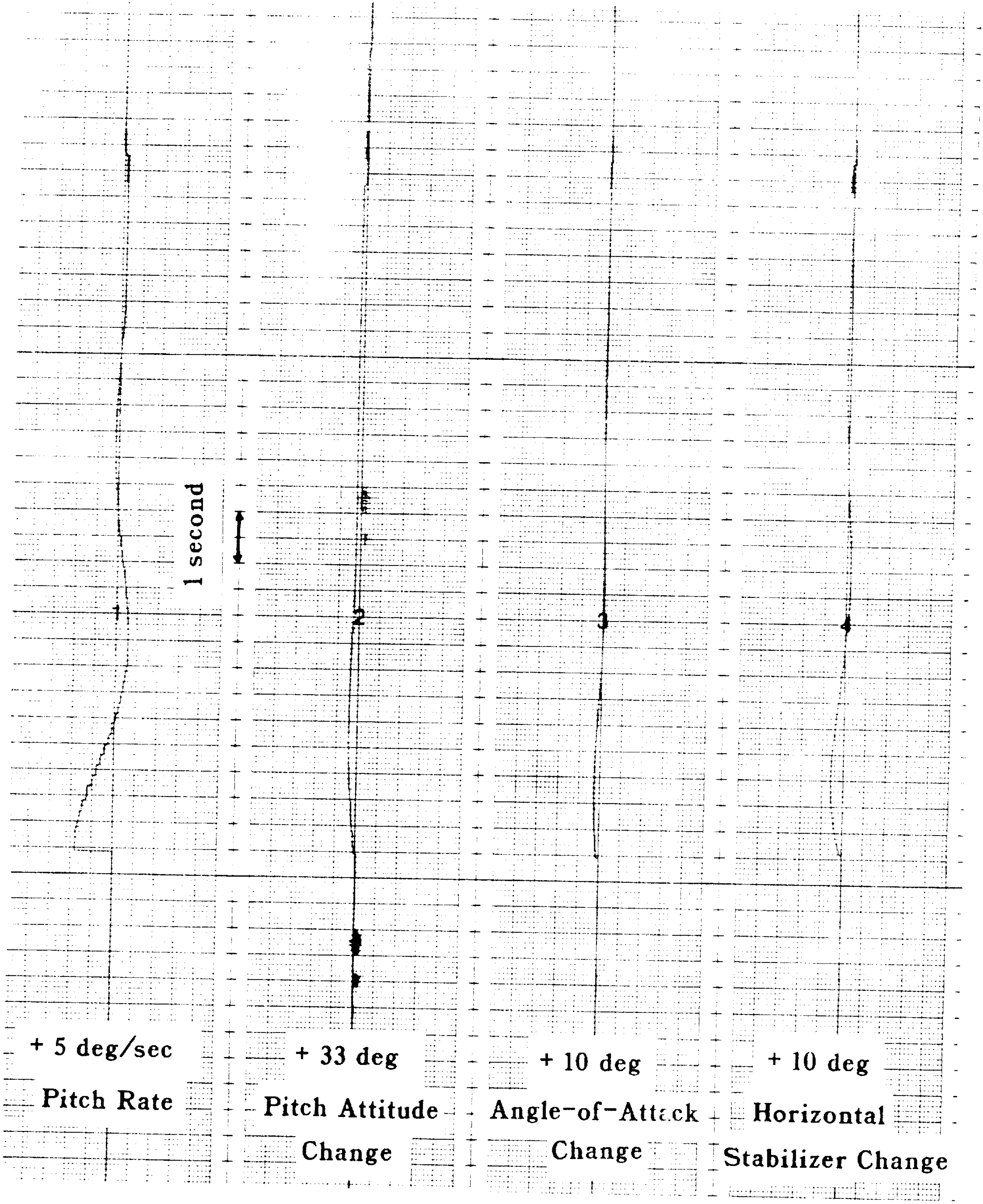


Figure 23. Stability Augmented Response without Pitch Rate

ORIGINAL PAGE IS
OF POOR QUALITY

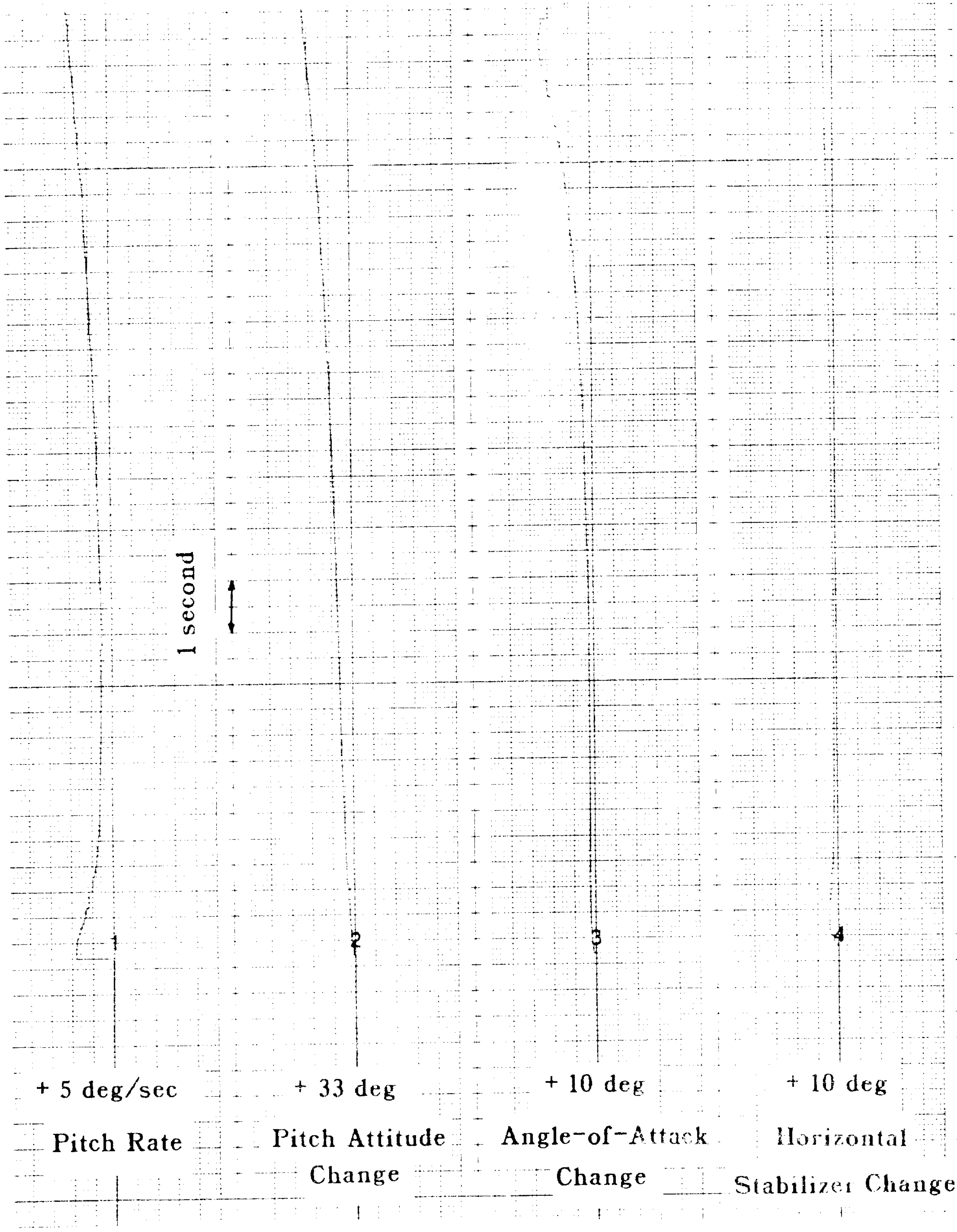


Figure 24. Stability Augmented Response without Angle-of-Attack

ORIGINAL PAGE IS
OF POOR QUALITY

5.0 CONCLUSIONS

Although the thrust of this effort was directed more toward the adaptation and demonstration of assurance methods in a subsetted development process, certain findings did result. The following general conclusions were derived or confirmed through the work described in this report:

- Digitally mechanized flight control systems are significantly more complex and difficult to validate than comparable analog systems

- An integrated, assurance-driven development methodology is particularly vital for fault-tolerant DFCS architectures

- Resultantly improved system/software structure greatly reduces the subject complexity, thereby facilitating validation

- Good structure also eases the impact of development changes, and reduces the number of validation test cases

- Automated multilevel testing can be effectively and beneficially performed in system simulators with modest additions

- Wideband software instrumentation and real-time test execution monitors are especially valuable additions for system simulator testing of critical DFCSs

- Use of the same execution monitor test cases and criteria for early-on analytical simulation and for all-up system simulation greatly extends confidence in the assurance process

- Multilevel test cases and criteria recapitulate and ultimately confirm the overall development process.

6.0 REFERENCES

1. DeFeo, P., D. Doane, and J. Saito: "An Integrated User-Oriented Laboratory for Verification of Digital Flight Control Systems - Features and Capabilities," NASA TM 84276, August 1982.

2. Barton, L.A., D.B. Mulcare, and R.J. LeBlanc: "Predicate/Transition Analysis of Redundant Channel Synchronization," AIAA Paper 85-6020CP, Computers in Aerospace Conference, October 1985.

3. Rang, E.R., et al.: "Specification and Analysis of Fault-Tolerant Systems," Honeywell Report 85SRC36, October 1985.

4. Sacks, I.J.: "Digraph Matrix Analysis," IEEE Transactions on Reliability, December 1985.

5. "System Design Analysis," FAA Advisory Circular AC 25.1309-1, 7 September 1982.

6. Ness, W.G., et al.: "Integrated Assurance Assessment of a Reconfigurable Digital Flight Control System," DOT/FAA/CT-82/154, April 1983.

7. Mulcare, D.B., and L.A. Barton: "N-Version Software Demonstration for Digital Flight controls," DOT/FAA/CT-86/33, April 1987.

8. Mulcare, D.B., L.E. Downing, and M.K. Smith: "Analytical Sensor Redundancy Assessment," DOT/FAA/CT-86/32, January 1988.

9. Boehm, B.W.: "Verifying and Validating Software Requirements and Specifications," IEEE Software, January 1984.

10. Reed, J.E., and E.M. Boothe: "Digital Avionics, Active Controls, and the FAA: Advanced Integrated Flights Systems (AIFS)," 2nd AIAA/IEEE Digital Avionics Systems Conference, November 1977.

11. Larsen, W.E., and A. Carro: "Digital Avionics Systems - Overview of FAA/NASA/Industry-Wide Briefing," 7th AIAA/IEEE Digital Avionics Systems Conference, December 1986.

12. Goldberg, J.: "General Concepts of Validation," in 'Validation Methods for Fault-Tolerant Avionics and Control Systems,' March 1979.

13. Wensley, J.H.: "Design for Validation," in 'Validation Methods for Fault-Tolerant Avionics and Control Systems,' March 1979.

14. Mulcare, D.B., W.G. Ness and R.M. Davis: "Analytical Design and Assurance of Digital Flight Control System Structure," AIAA Journal of Guidance, Dynamics, and Control, May-June 1984.

15. "General Specification for Design, Installation and Test of Piloted Aircraft Flight Control Systems," MIL-F-9490D, U.S. Air Force, 6 June 1975.

16. McCabe, T.J.: "A Complexity Metric," IEEE Transactions on Software Engineering, December 1976.

17. Benson, J.W.: "Development and Implementation of the Real-Time Six Degree of Freedom Airplane Simulation for the Reconfigurable Digital Flight Control System," Prepared under Contract NAS2-10270, December 1981.

*U.S. GOVERNMENT PRINTING OFFICE: 1988--505-068/80130

NASA
FORMAL
REPORT